AN ONTOLOGY-BASED BIM EXPERT SYSTEM FOR TEMPORAL AND SPATIAL CONSTRUCTION PLANNING

Dissertation

submitted to and approved by the

Department of Architecture, Civil Engineering and Environmental Sciences
University of Braunschweig – Institute of Technology

and the

Department of Civil and Environmental Engineering
University of Florence

in candidacy for the degree of a

Doktor-Ingenieur (Dr.-Ing.) /

Dottore di Ricerca in Civil and Environmental Engineering*)

by

Vito Getuli
born 01/12/1987
from Atripalda, Avellino

Submitted on	25 August, 2017
Oral examination on	9 November, 2017
Professorial advisors	Prof. (Klaus Thiele)
	Prof. (Pietro Capone)

2018

*) Either the German or the Italian form of the title may be used.

I dedicate this thesis

to my sister ***Clelia***

for her unconditional love

Abstract

The effective realization of building construction projects is closely linked to the construction activities planning process that should consider the workspaces availability in site, according to their dynamic nature. Poor schedules estimate results in congested site areas, wasteful material movements, accidents and decline of productivity. In this context, in the past couple of decades many research efforts have been spent in BIM which represents the process of preparation and use of a computer-generated Building Information Model (BIM) even if effective integrated methodologies and models to assist construction scheduling are still missing in the field.

This PhD thesis proposes an **Expert-System**, driven by a construction spatial scheduling algorithm, which aims to identify the shortest completion sequence of a given Building Information Model, considering the on-site temporal-space allocation of workspaces. It is supported by an *ontology-based system architecture* (for semantic modelling the construction process knowledge) integrated with a rule-based artificial intelligence (for workspaces management and generation of schedules).

A complete ontology, still missing in literature, constitutes the system's **Knowledge-Base** (KB). It was developed to formally represent the construction site entities in the form of classes, relationships and properties. A multi-domain modelling approach which integrates four sub-ontologies with the use of BIM data, to support generation of schedules, was used. These ontologies are (1) *scheduling ontology* that maps the necessary components to specify the scheduling task (2) *space ontology* that contains workspaces requirements in terms of geometries, locations and interactions (3) *products ontology* that describes functional, geometrical and topological information of the building objects for the scheduling purpose (4) *time ontology* that describes temporal properties of site entities in their evolution across time. Such a KB was rendered into a *Protégé's script* (ontology editing environment) in order to convert it in machine-readable language (i.e., *Web Ontology Language –OWL-*).

Furthermore, four automated **Reasoning Mechanisms** –scripts- were developed and incorporated in the proposed model architecture: (i) an algorithm to define the on-site workspaces configuration pattern based on a space syntax analysis, (ii) an algorithm to automatically model workspaces geometries with minimized input work, (iii) a workspaces conflicts checking process within a BIM simulation environment and (iv) a rule-engine which contains the ontology-based scheduling rules to deduce the shortest construction sequence and solve the identified conflicts manipulating the KB itself.

A **validation test** was conducted on a BIM-based project of an industrial building. A full building model, including 98 building items and 611 workspaces, allocated by means of (i) and modelled with (ii), was produced and a construction sequence of 36 construction days was suggested by the system. Moreover, 118 workspaces conflicts were identified (iii) and automatically solved by using the planning rules included in the rule-engine as it was visually verified simulating such a construction sequence within a 4D-BIM environment.

This prototype introduces significant automation in existing construction scheduling methodologies and it can be considered a precursor model in developing BIM-based intelligent system architectures for construction spatial planning.

Kurzfassung

Eine erfolgreiche Umsetzung eines Gebäudeprojektes ist von der Planung der Montage auf der Baustelle abhängig. Im letzten Jahrzehnt wurden zahlreiche wissenschaftliche Projekte zur Montageplanung unter Verwendung eines Computermodells im Rahmen des **Building Information Modelling** (BIM) durchgeführt. Momentan fehlt aber noch ein Modell, das auch den Prozess selber auf der Baustelle integriert.

In der vorliegenden Arbeit wird ein **Expertensystem** mit dem Ziel der Findung einer optimalen Montagefolge vorgestellt. Das Expertensystem basiert auf BIM und berücksichtigt die räumliche und zeitliche Interaktion der Arbeitsabläufe auf der Baustelle. Die entwickelte Methode stützt sich auf einer Ontologie-basierten Architektur, die in einer Regel-basierten künstlichen Intelligenz integriert ist. Dabei wird ein neues Objekt in das BIM Modell eingefügt, das den Raumbedarf einer Montagetätigkeit beschreibt. Dies kann beispielsweise ein erforderlicher Freiraum für einen Mobilkran sein oder ein bei der Montage nicht betretbarer Sicherheitsbereich.

Die Wissensbasis (**Knowledge-Base**) des Expertensystems besteht aus vier Ontologien, die nötig sind um das Wesen der Baustelle darzustellen:

(1) Ontologie der Montageablaufs, die den technischen Ablauf der Aktivitäten bestimmt;
(2) Ontologie der baulichen Räume, die den räumlichen Bedarf berücksichtigt;
(3) Ontologie der Elemente des Gebäudes, welche die geometrischen und funktionalen Gebäudeelemente beschreibt, um Arbeitsprozesse zu bestimmen;
(4) Ontologie der Zeit, welche die Reihenfolge der Bauelemente vorgibt.

Die Wissensbasis ist mit einem Protégé-Skript als Ontologie-Editor entwickelt worden, für einen Compiler der Web Ontology Language (OWL). Danach wurde die Wissensbasis mit vier Algorithmen verknüpft:

(i) Ein Algorithmus, der den Arbeitsraum definiert;
(ii) Ein Algorithmus, der die Geometrien der Arbeitsräume modelliert;
(iii) Ein Kontrollprozess, der die Konfliktstellen des Arbeitsraum identifiziert;
(iv) Ein Optimierungs-Prozess, der den kürzesten Arbeitsprozess ermittelt.

Zur **Validierung** wurde ein Industriegebäude mit 98 Elementen verwendet. Das Expertensystem hatte 611 Arbeitsräume errechnet und eine geschätzte Bauzeit von 36 Tagen. Das Expertensystem identifizierte 118 Konfliktstellen und entwickelte jeweils Lösungen. Das Ergebnis wurde mit Hilfe einer 4D-BIM Umgebung visualisiert. Die Validierung fand mittels Plausibilitätskontrolle statt. Da vorgestellte Expertensystem ist ein Prototyp, der einen Beitrag zur Entwicklung automatischer und intelligenter Programmierungen für den Montageablauf unter Verwendung von BIM leistet.

Acknowledgements

First and foremost, I want to express my sincere gratitude to my tutors, Professors Klaus Thiele and Pietro Capone. It has been an honor to be their student.

I am very grateful to Professor Thiele; he has always been capable of seeing things clearly in the way only the best supervisors know how. He taught me to focus my energies particularly when I was lost in my research ideas. I still remember when he told me: "Vito you don't have ten years of research but only three". I am also very thankful for his idea to introduce BIM in my research during my period in Braunschweig. Now BIM is for me not only an exciting research field but a very passion.

Prof. Capone is someone you will instantly love. He's the funniest advisor and one of the smartest people I know. I hope that I could be as lively, enthusiastic, and energetic as him and to someday be able to guide an unaware young student, like me four years ago, as well as he has done. I have learnt extensively from him, including how to raise new possibilities and how to regard an old question from a new perspective. He has been supportive and has given me the freedom to purse various projects without objection, trusting in me and funding to make my PhD experience productive and stimulating. After all, whatever I am today, I owe to him. I feel privileged to be associated with a person like him during my life.

I am also very grateful to my external reviewers, Prof. Angelo Ciribini e Prof. Shabtai Isaac for their scientific advice and knowledge and the insightful discussions and suggestions.

This years were made enjoyable also due to my colleagues, now friends. I am grateful for our memorable trips. Pina is a wonderful and generous friend and I admire her positive outlook and her ability to smile despite the situation. The music of Andrea has given me many headaches as well as moments of self-reflection. I shared moments of crisis with Federica; she is always the most worried but always had some reassurance. Emanuele is a true friend; our research discussions have always been stimulating. Finally, my roommate Dario, my first personal trainer with an admirable sense of duty.

I also thank Professor Kindinis, a nice and helpful person who has been pretty supportive during my research period in Paris at the '*Institut de Recherche en Constructibilité*'.

I'm also glad to have worked as supervisor with many wonderful undergraduate students; especially Tommaso, Andrea, Federico, Giulia, Giorgio, Simona, Francesco, Rossella and Francesca. They are allowed me to make many tests that have supported my research work.

My special words of thanks should also go to Silvia, a dreamy researcher. She is always ready to listen my proposals of joint research ventures as well as to share her knowledge freely with me.

I feel a deep sense of gratitude for my mother and father. They have sacrificed their lives for my sister and myself and provided unconditional love and care; I know I always have them to count on when times are rough.

Special thanks to Eleonora and Clelia.

One of the best conclusion from these past six years is finding my soul-mate. Eleonora is the only person who can appreciate my quirkiness and frenzy. She has been a true and charming supporter and has unconditionally loved me during my good and bad times. She has been non-judgmental of me having faith in me and my intellect even when I felt like digging hole. The past four years have not been an easy ride, both academically and personally. I truly thank Eleonora for sticking by my side, even when I was irritable and tired. I believe that we both learned a lot about life and strengthened our commitment and determination to live our life together to the fullest.

I owe my deepest gratitude to my lovely sister Clelia for her eternal support and understanding of my goals, aspirations as well as my fragility. Her infallible love and support have always been my strength. She is holding my hand ever since we were kids. Her sentimental and intellectual honesty will remain my inspiration throughout my life. Without her help, I would not have been able to complete much of what I have done and become who I am. It would be thankless if I thank Clelia in these few words and for this reason I dedicate this thesis to her.

Vito Getuli
Università degli Studi di Firenze
Technische Universität Braunschweig
November 2018

Index of Contents

List of figures

List of abbreviations

- **AEC** Architecture, Engineering and Construction industry
- **BIM** Building Information Model
- **BIMs** Buildings Information Models
- **2D** Two-Dimensional
- **3D** Three-Dimensional
- **4D** Four-Dimensional
- **IFC** Industry Foundation Classes
- **LoD** Level of Development
- **LoI** Level of Information
- **VC** Virtual Construction
- **VR** Virtual Reality
- **ES** Expert System
- **ESs** Expert Systems
- **KB** Knowledge Base
- **IE** Inference Engine
- **UI** User Interface
- **RE** Rule Engine
- **OWL** Web Ontology Language
- **SWRL** Semantic Web Rule Language
- **JHA** Job Hazard Analysis
- **GIS** Geographic Information System

Glossary

A

Activity A process in a project that consumes time and also usually consumes or uses other resources (e.g. people, money, materials and equipment). An activity is the smallest unit of work on a **Schedule** but, depending upon the hierarchy or level of detail of the schedule, may be divisible into smaller or more detailed activities.

Activity Duration The time calculated or estimated to carry out an activity, generally taking into account a specific level of resources, constraints and methods of working.

Activity- Oriented Scheduling The method of developing a schedule that determines the sequence and timing of activities based on the logical work process only and does not take account of any potential limitations of resources.

B

Bar A line on a bar chart that represents the timing and duration of an activity.

Bar Chart A graphical chart on which activities are represented as bars drawn to a common time scale. Typically, a date scale is drawn across the top of the page and a list of activities down the left hand side of the page. Activity timing and durations are represented by horizontal bars.

Baseline Schedule A fixed or record schedule against which current or future activity is referenced. Often taken to mean the first or original plan but can be reset (for instance, following a change to the project scope), at which point the reset schedule becomes the (new) baseline schedule.

BIM See **Building Information Model**

Building Information Model is a virtual representation of a building, potentially containing all the information required to construct the building, using computers and software. The term generally refers both to the model(s) representing the physical characteristics of the project and to all the information contained in and attached to components of these models. When BIM is used in a sentence, it will depend on the context whether it means building information model or building information modeling. A BIM may include any of or all the 2D, 3D, 4D (time element—scheduling), 5D (cost information) representations of a project.

Building Information Modelling is the act of creating and/or using a BIM

C

Calendar A list of time intervals during which activities can be worked and/or resources used. Typical data includes working days/non working days, start and finish times for shifts. Each activity and/or resource will have a calendar attached to it.

Component The word component may refer to an element in a 3D model, or it may also indicate an individual part of a BIM, e.g., the mechanical model or the structural model. It will be necessary to derive the specific meaning from the context.

Constructability This term refers to the analysis of the ability to construct. In the early phases of a project, such analysis can provide valuable input for the practicality of the assembly process of a project. Constructability analysis can take place on various scales, depending on the phase of the project and the level of detail available about the construction process.

Construction Project This is synonymous with building project, and it refers to the planning, preparation, and construction of a building. Projects typically are performed by individuals who use methods to achieve certain results.

D

Duration The estimated or actual time required for the completion of an activity, or a group of activities, based upon a particular resource allocation and method of working.

Dimensionality - 2D, 3D, 4D, 5D The common convention referring to the "geometric dimensions of some physical or abstract system" (Webster's *New World College Dictionary*), where 2D space is a flat plane; 3D space is three-dimensional space, e.g., length, width, and height; 4D space adds time as a dimension; 5D space will generally refer to cost.

E

Earliest Finish The earliest time that an activity, or a group of activities, can finish within the constraints, resources and logic of the network.

Earliest Start The earliest time that an activity, or a group of activities, can start within the constraints, resources and logic of the network.

F

4D Planning and Scheduling The integration of schedule and graphics to produce a time-based visualization of the development of a project. Predominantly carried out by linking Project Management Software with graphics/drawing software though integrated software is now available.

Field The field is a term usually referring to the physical construction site when it is used in a discussion of construction topics.

G

Gantt Chart A Gantt chart is a graphical representation of the duration of activities against the progression of time and is a particular type of Bar Chart though used as a synonym for bar charts in general.

H

Hazard The potential to cause harm, including ill health and injury; damage to assets, products or the environment; production losses or increased liabilities.

I

Industry Foundation Classes IFC means *industry foundation class*; it is a term coined by the International Alliance for Interoperability. The IFC is a standard file format for 3D models that will permit information to be exchanged among all models that can be translated into this file format. It is an attempt to bring about standards for a common language between the various model authoring and analyzing software tools.

Interoperability *Interoperability* refers to the ability of different file formats to be integrated with one another and transfer relevant information among one another.

J

Job Safety Analysis (JSA) A procedure used to review job methods and uncover hazards that may have been overlooked in the layout or design of the equipment, tools, processes or work areas; that may have developed after work started; and that may have resulted from changes in work procedures or personnel (ECI, 2013).

M

Milestone A zero duration activity used to identify or highlight key points of Events in the project. Milestones are often used to identify the start or completion of sections of the project and therefore useful for Monitoring performance.

Monitoring The recording, analysing and reporting of project performance and comparing it to the **Baseline Schedule**.

O

Ontology In the context of computer and information sciences, an ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members). The definitions of the representational primitives include information about their meaning and constraints on their logically consistent application. In the context of database systems, ontology can be viewed as a level of abstraction of data models, analogous to hierarchical and relational models, but intended for modeling knowledge about individuals, their attributes, and their relationships to other individuals.

Object-Based Model The use of objects in 3D models renders them more usable and efficient in the BIM process. An object generally represents a physical entity (although nonphysical entities, e.g., events such as inspections, can also be represented by objects) in the project and is able to contain information relevant to the project. Objects are often composed of many parts that would be much more burdensome to the project model if treated as separate parts.

P

Planning The process of preparing for the commitment of resources in the most effective fashion. It aims to produce a workable schedule that will achieve project goals and serve as a standard against which actual progress can be measured. It defines: (1) What should be done (activities); (2) How should activities be performed (methods); (3) Who should perform each activity and with what means (resources); (4) When activities should be performed (sequence and timing)

Planner A member of the project team responsible for planning, scheduling and monitoring the progress of the project. Often used as a synonym for Scheduler.

Progress The measurement of the completeness of an activity or a group of activities or the project as a whole.

Project A project may be defined as a temporary endeavour undertaken to create a unique product. Projects are performed by people, constrained by limited resources, planned, executed and controlled. A typical construction project includes construction work incorporating planning, design, management or any other works involved until the end of the construction phase – that is it includes construction, alteration, conversion, maintenance, fitting out, commissioning, renovation, repair, upkeep, redecoration, decommissioning, demolition or dismantling. A full definition is available in the Temporary and Mobile Construction Sites Directive (92/57/EEC) and relevant national legislation.

Project Schedule The term Project Schedule may be interpreted in two ways. (A) The project activities and milestones and durations and planned sequence and timing; (B) The physical document (**Bar Chart**), for instance, that illustrates and communicates the aforementioned.

Parametric A parametric object or component is an object (or component) that permits a choice of values for defined parameters. A parameter is a variable value (as in a mathematical equation) that, when it changes, gives a different but related characteristic to the original object. An example is a steel beam in a 3D model that can have the size of the beam as one of its parameters. This means that the specific beam in the model needs to have its size specified, and it will thus reflect its physical size and weight accurately in the model. The chosen values for the parameters generate parametric information. In the case of the steel beam, the size of the beam implies a variety of information that will be determined by its size, e.g., its width, thickness, total weight (resulting from the length), etc.

R

Reschedule A calculation performed on the tasks and links to ensure that the project is completed in the minimum possible time within the logical and imposed constraints of the plan and any progress that might have been achieved.

Resource Any goods or services required to complete the work of an activity. For example, labour, materials, plant and money.

Resource-Oriented Scheduling The method of developing a schedule that determines the sequence and timing of activities based on the logical work process and the availability and limitation of resources. Generally, this involves estimating activity durations based on available resources and the introduction of Resource Links to stimulate the transfer of resources from one activity to another.

S

Schedule (1) The timetable for a project. Showing how **Activities** and **Milestones** are planned to be carried out over a period of time; (2) The physical document for communicating the **Plan**, especially timing and sequence.

T

Taxonomy It is the practice and science of classification. A taxonomy, or taxonomic scheme, is a particular classification. Specifically A Data Taxonomy is a defined classification of terms, organized hierarchically into any number of levels of category and sub-category as required, and to serve a given purpose.

V

Virtual Construction Virtual means *not real*, and this refers to the processes taking place in the computer. Virtual construction is a term used by CIFE and Graphisoft to describe the use of a 3D computer model to simulate not only the design of a structure but also its assembly, the construction process. Virtual construction will likely include the analysis of the construction schedule.

Chapter 1 Introduction

1.1. Motivation and problem statement

Whatever the type of building project under consideration, there is a need to plan. Whatever the type of building project under consideration, there is a need to produce a schedule. Construction activities *planning and scheduling*[1] are widely considered challenging activities in managing construction projects due to the numbers of variables and choices construction managers should consider especially resulting from the dynamic nature of construction site in ways that fixed industrial facilities do not: work tasks are transient, the physical and temporal structure of equipment, site objects and workspaces change constantly, and construction site are exposed to the environment and changes in weather.
Considering its nature, which consists of different activities in limited area, it can be stated that '*workspace*' is the leading factor that is frequently overlooked in scheduling construction activities, resulting in congested site areas, wasteful material movements and equipment relocation which cause decline of productivity and occurrence of accidents when it is too late for rescheduling activities (Hegazy, 1999).

Therefore, construction schedules which do not consider such a factor can easily result in large construction cost increases or delays or safety hazards, owing to inappropriate decisions concerning work tasks, required resources and particularly workspaces planning (Sacks, 2009).

In this regard, it is just as important as challenging to consider workspace availability in generating construction schedules of a given building project due to the complexity of modeling and planning workspaces as well as spatiotemporal relations among workspaces themselves, activities and building objects they refer to and, if needed, by even using semi-

[1] Even though the two terms *planning* and *scheduling* are often used interchangeably they are separate tasks in construction management theories and practices. Due to the fact that the presented research involves both of them, the Chartered Institute of Building (CIOB), definition is useful: "Project planning is an experienced-based art, a group process requiring contribution from all affected parties for its success. Scheduling is the science of using mathematical calculations and logic to predict when and where work is to be carried out in an efficient and time-effective sequence" (CIOB, 2011). It is possible to consider *planning*, as the activity which aims to produce a workable schedule in order to define (1) What should be done (construction activities), (2) How should activities be performed (construction methods), (3) Who should perform each activity and with what means (resources), (4) When activities should be performed (sequence and timing). And instead, *scheduling* can be considered the particular planning duty which aims to determine when the activities should be performed (sequence and timing), utilizing methods such as Bar Charts and Networks or new developed methods (Bordoli, 2014).

automatic models.

In this perspective, several approaches and models have been proposed in the last decade to coordinate construction activities (schedules) with site organization and layout sometimes attempting to introduce time-space conflicts analysis in order to describe situations in which two activities overlap in time and their workspaces are interfered during the time overlap (Cheng, 2013). Visualization techniques such as VR (Virtual Reality)[2], 3D and 4D Graphics supported the development of such models.

More recently, a new research trend of utilizing Building Information Modelling (BIM) and BIM-related technologies and processes to assist construction scheduling is arising. It represents the processes of preparation and use of a computer-generated Building Information Model (Azhar, 2008), the so-called BIM, which is data-rich parametric digital representation of a building, from which relevant data, needed to support planning and scheduling of construction activities, could be extracted and analyzed (Ernstrom, 2006). But, in spite of its growing implementation in the last decade, the use of BIM to improve design and planning of construction process has involved many efforts focusing on its technological aspects (tools) rather than on the integration with planning and scheduling models.

Therefore, before leaving the subject, it could be said that first simulation and then planning of construction activities with reference to their workspace requirements are crucial to guarantee a high *constructability* level of a given building project (Gambatese, 2005).

In this context, the reviewed studies exhibited the following open issues:

(1) unlike manufacturing, construction activities are not repetitive, each building project being different in layout, construction methods, materials, and so forth. Therefore, ready algorithmic solutions are non-applicable;

(2) construction environment is full of uncertainties due to workers, equipment, material availability and in many construction situations there is not enough time to develop an in-depth analysis able to produce detailed construction scheduling considering the spatial factor. Expert systems can provide a strategic guidance in such situation;

(3) a scheduling model which considers workspace availability and demand could support planners in producing shorter schedules that feature concurrent execution of overlapping activities without resulting in spatial conflicts but it is still missing (Said, 2016). A number of complex mathematical models have been proposed in literature but they are not able to

[2] *Virtual Reality* (VR) is an important approach in current BIM (Building Information Modelling) research and vice versa (Gu, 2010). VR is a virtual system that consists of a computer capable of real-time animation, controlled by using a group of equipment for simulating physical presence in places in the real world (Steuer, 1993). Until now, it has been used to provide a 3Dimensional, virtual and interactive computer-generated environment for training site workers to become aware of identified on-site safety risks (Guo, 2012) and develop both strategies and measures of potential hazards by simulating dangerous scenarios (Wang et al., 2014).

manage a building project in its entirety and they lack flexibility if changes in dimensions and relationships of workspaces and building objects occur;

(4) the integration in a unique model of human planner, BIM (Building Information Model) and automation in planning and scheduling processes is still missing;

(5) lastly, traditional construction scheduling techniques such as *Gantt charts* and *Network Diagrams* are inadequate for managing site workspace conditions and their topological relationships mainly due to their lack of spatial representation (Moon, 2014).

In the light of the foregoing, it can be concluded that a flexible, automated and integrated *holistic* model, able to work within a BIM compliant environment, to predict the shortest construction schedule, especially considering workspaces requirements of construction activities, is still missing. Such a model would effectively support the *construction process simulation* and it should surely consider three factors:

(a) introduction of a workspaces planning model,
(b) generation, detection and resolution of site workspace conflicts (Kassem, 2015),
(c) integration of construction managers' experience in planning,
(d) Building Information Models as created by compliant tools should be used as data source for the schedule generation (Mikulakova, 2010).

By embracing these open issues, the PhD thesis therefore presents a new *BIM-compliant spatial scheduling algorithm*, which has been computerized by using an ontology-based expert system that explicitly considers the workspace component.

1.2. Objectives and solutions

Tentative objectives of this PhD thesis are summarized as follows:

(i) Elaboration of a *knowledge Base* (KB) capturing all those *entities* able to depict and simulate construction activities mainly in terms of space temporal requirements, as well as their relationships with building objects and construction methods. Such a KB would contribute to stand out the current gaps in a fragmented literature and support the generation of *construction schedules*.

(ii) Development of a system engineering able to computerize a new *spatial scheduling algorithm* within a BIM-compliant environment, aiming to automate both site workspaces planning and site activities scheduling, looking to obtain the shortest possible total construction duration of a given Building Information Model. All this avoiding the fragmented use of methods and tools presented in literature until now.
In this perspective, investigations have been carried out in other research fields (manufacturing management, aviation and crisis-action logistics planning) in order to

understand how scheduling systems are managed within a more advanced sector compared to the AEC[3] one. However, in view of the complexity of this issue, the model might lay the groundwork for future developments and extensions.

In the light of the aforementioned objects and based on the results of the review and analysis of the current knowledge and models (*Chapter 2*), this research proposes the development of an *expert system*, operating by means of an *ontology-based architecture* in communication with a *BIM-based modelling and simulation environment*. It is called **`OnSITEsimu`**.

The provided model may be specified as follows in terms of proposed objects and given solutions:

1) Objective 1) The system should capture and manage knowledge on the construction site entities and their connection among construction methods, resources, workspaces requirements as well as their topological interaction in the typical site dynamic progression. Reasoning mechanism should be integrated.

Solution 1) For the solution of the given object, we propose to formally express site entities and conditions by using *ontologies*. The reasons are threefold.

- *First*, concepts and their semantic relationships in depicting the construction process can be represented in the form of *classes*, *relationships* and *properties* in an intuitive way.
- *Second*, ontologies support reasoning mechanisms by means of a specific computational language that allows for a standardized way to express the scheduling problem.
- *Finally*, ontologies from different sources can be connected if they share the same language.

2) Objective 2) The system should provide a means to formalize a spatial scheduling algorithm, by using knowledge before mentioned in form of ontologies, which is intended to automatically extract the shortest construction sequence considering the temporal-space allocation of resources within construction site so as to permit the development of automated design assistant.

Full automation is no longer wanted but integration of experiences from the user is investigated. Moreover, automatic reasoning based of the scheduling strategy should be integrated. In this regard, in the realm of artificial intelligence, numerous methods address the general problem of planning, even if not specifically in the context of construction.

Solution 2) We propose, a *spatial scheduling algorithm* to identify the progress direction as well as start and end time of each activity to obtain the shortest possible total

[3] Architecture, Engineering and Construction (AEC) Industry: the sector of the construction industry that provides researches and services on the architectural design, engineering design and construction.

construction duration. The algorithm is created as a greedy procedure that utilizes a set of spatial heuristics and predefined rules to plan the workspaces configuration patterns and schedule the activities in a chronological manner. The algorithm is implemented in a computerized ontology-based Expert System (ES) just as explained in *Chapter 3*.

The choice to use an ES as model to get the scheduling algorithm operational is due to its functional components that we think would go great with our mutual goals:

- a *knowledge base* which includes factual knowledge (i.e., a model which describes the construction process across time),
- a *control mechanism* (i.e., an inference engine able to reproduce the scheduling strategy)
- finally, information about a *particular domain* (i.e., data collection from the given BIM and user's experience)

3) **Object 3**) The review studies consider 2D, 3D and 4D CAD-based environment, in which workspaces –volumes- are simulated or rather manually generated through mark-up. This is time-consuming and unmanageable, due to the number of needed workspaces in a given construction process with hundreds of activities. Furthermore, automation in workspaces modelling and detection of spatial-temporal conflicts is a crucial research object. Moreover, changes in building components or construction methods need to produce changes in the construction sequence as suggested by the system itself.

Solution 3) The integration of the proposed model with a 4D BIM environment (IFC compliant) is pursued in order to automate the process of building data acquisition as much as possible. The integration is pursued for the following reasons:

- to automatically provide the model with information about dimensions, allocations and structural connections between building components;
- to automate the workspace generation and detection of workspaces conflicts.
- to simulate the construction sequence by means of a 4D BIM-based simulation environment.

In the next paragraph the main blocks regarding the research development are presented in their natural progression with reference to the relative chapters.

1.3. Methodology and development process

The methodology adopted to achieve the aforementioned objectives is schematically summarized in the flowchart in Figure 1-1 and specifies as follows.

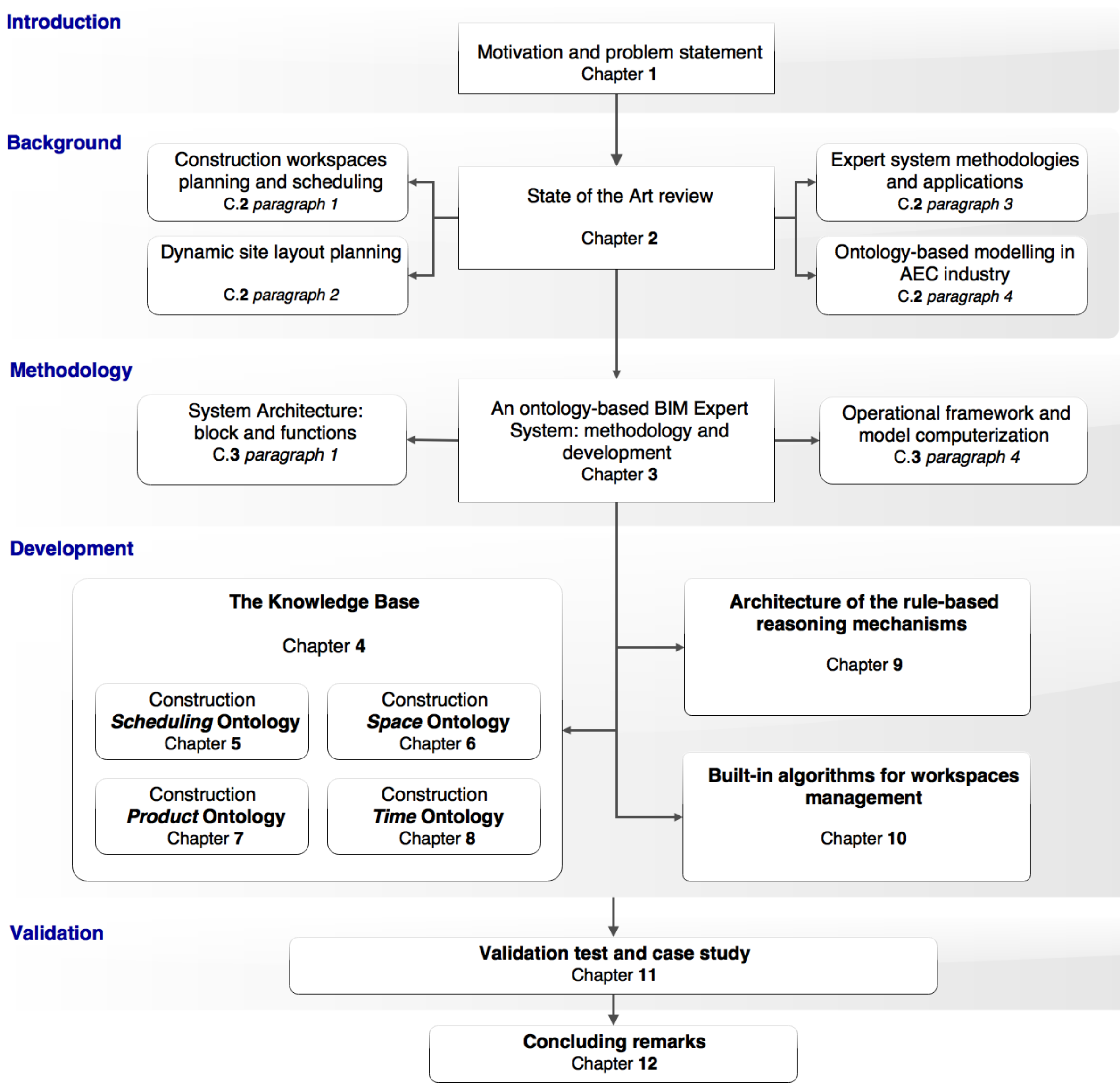


Figure 1-1 Development process of the *PhD thesis*: main blocks and related chapters

After describing in *Chapter 1* motivations and open issues which led to propose this research development, in *Chapter 2* we review related studies making a comparison with a list of research requirements in order to storage the knowledge on the addressed topics. This is considered fundamental to develop of a new construction spatial scheduling model. The review of four main topics is presented: (1) Construction workspaces planning and scheduling, (2) Dynamic Site Layout Planning, (3) Expert system methodologies and applications, (4) Ontology-based modelling in AEC Industry.

In *Chapter 3* the architecture of the proposed system is presented. After describing the main development issues, its operational framework and computerization are described giving a complete overview of specific points that have subsequently been addressed.

The *Chapter 4* presents the core of the expert system: the *Knowledge-Base*. Giving specifications on the ontology-based modelling approach and its development architecture the sub-ontologies, which drive the system, are presented. Specifications in terms of modelling entities, topological relationships and properties sets of construction site entities are described, as well as the *scripts* that make such a KB machine-readable.

- *Chapter 5* provides specification on the proposed *scheduling-model*: the so-called Scheduling-Ontology.
- *Chapter 6* deals with the proposed ontological structure to represent workspace requirements within the construction process. A complete *workspace-model* which has been formalized in the Workspace-Ontology.
- *Chapter 7* describes the framework of the building-model, once again in the form of ontology that works as a bridge between the proposed system and BIM-based internal standard.
- *Chapter 8* introduces the ontology-based *time management* able to support the system in generating construction schedules.

Chapter 9 explains the introduction of Artificial Intelligence in the system architecture. It starts describing how the rule-engine works to reach a solution and then it presents in details the rules-set, giving specifications on their syntax with reference to the aforementioned ontologies.

To conclude the reasoning mechanisms, *Chapter 10* presents the built-in algorithm which drives the system to define a spatial configuration pattern of workspaces in site by using the Space Syntax Analysis.

Lastly, the proposed model has been validated by using a simplified building model of an industrial building and the results are presented and interpreted in *Chapter 11*.

Chapter 2 Background

2.1. Construction workspaces management

One of the most important factors able to affect the efficient and safe delivery of construction projects is the site workspaces availability (Dawood, 2005). Workers, equipment, temporary facilities have different space requirements and they compete with each other for space usage throughout the entire life of a project (Cai, 2014). Furthermore, workspaces interact with each other dynamically, their locations and volumes change in three dimensions and across-time according to a specific schedule information. In this context, traditional construction scheduling techniques such as Gantt charts and network diagrams are inadequate for managing site workspaces, mainly, due to their lack of spatial representation (Chau, 2004), (Dawood, 2006), (Dawood, 2009), (Moon et al., 2014). For the aforementioned reasons, incorporating workspace consideration from the spatial and temporal perspective in construction planning plays a pivotal role. Several expressions have been used to describe this process to involve workspace management including 'spatial modelling', 'execution space analysis', schedule-workspace management', 'workspace planning' and 'time-space analysis'. According to Kassem (2015), the definition *Construction Workspace Management* includes three elements:

(a) generation and allocation of workspaces,

(b) detection of congestion and spatial temporal conflicts and

(c) resolution of identified conflicts.

Since these components are considered in the research, a deeper research investigation has been carried out.

(**a**) Most of early studies, between 1992 and 2006, focused on ***generation of workspaces*** and ***detection of conflicts***.

Thabet (1994) proposed a methodology in which workspace are marked up in an CAD environment and then are broken down into work blocks and linked with related activities. By using a 'space capacity factor' congestion is detected and rule-based strategy is used to solve the congestion. Sanvido et al. (1995) presented a scheduling model that incorporates workspace constraints in the scheduling of repetitive work in multistorey buildings. Their model proposed a method to define and quantify several workspace parameters such as physical space demand

and surrounding space demand. Once again Riley and Sanvido (1997) presented a space planning method in the case of a multistorey building. The space conflict was reviewed empirically on 2D plan view by defining space type and identifying an execution location orders for each considered space. Guo (2002) analyzed workspace overlapping introducing the values: Interference Duration Percentage (IDP) and Interference Space Percentage (ISP) in 2D floor plan.

Akinci and Fischer (2000) presented a methodology for the generic generation and allocation of workspaces to activities that considers the construction methods in use. The methodology includes an ontology for the capture of spatial requirements that is implemented in ad-hoc 4D CAD environment (i.e., 4D SpaceGen) for the automatic generation of workspaces.

The so-called Geometry-based Process Model (GPM) was developed by Akbas (2004), it models the conversions in construction processes as sequences of crews acting on geometric work locations. It uses a simple process description: work locations are processed by crews.

Dawood et al. (2005) applied entity-based 4D CAD technology for detecting workspace congestion to help identify potential safety hazards on-site using critical space-time analysis (CSA) in 4D visualization. The proposed CSA associates certain visual features for workspace planning with the workspace competition. The PECASO (Patterns Execution and Critical Analysis of Site-space Organization) prototype was developed to encapsulate and evaluate the outcome of the CSA.

Song and Chua (2005) established methodologies for a system modeling of temporal 3D space, using COmponent State Network CEntric Model (COSCEM) aiming to present a platform for integrating project information including product, process, space and intermediate function.

Thomas et al. (2006) considered a real case study to analyze the effects of documentation of workspace conflicts and labor productivity in order to minimize the workspace congestion in the case of a multistorey building project.

Moon et al. (2009) proposed an integrated approach in which workspaces are generated using the bounding volume of each individual objects and then are linked with schedule activities in a 3D CAD Environment.

Winch and North (2006) suggested a Critical Space Method (CSA) in order to solve the construction space scheduling problem and developed *AreaMan* and *SpaceMan* for the CSA system that supports managers to analyze the spatial overloading between work execution spaces.

Chavada et al. (2012) developed approaches and a prototype for visually analyzing congestion of activity execution workspaces (AEWs) with the severity of congestions (CgS) by calculating the conflicting volume between workspaces in nD CCIR (Dawood and Mallasi, 2006), (Mallasi, 2006).

The so-called method ‘Spatial Network’ in which workspace requirements are considered only at a relatively high level of detail for resources such as laborers and building objects was proposed by Bargstädt and Elmahdi (2010).

Cai and Su (2014) presented a lifecycle approach to the modelling and planning of construction workspaces taking into account the evolution pattern of activities' space requirements. The aim was reached using an object-oriented structure of workspace with both geometric and temporal attributes. This research, implemented in an ad-hoc workspace modelling and planning environment -independent from commercial scheduling and design platform-.

Jongeling et al. (2008) took into account the distance between the different types of work as the main relevant factor which affects a safe and a productive work execution, by manually extracting 4D spatial content from 4D CAD models.

Recently Kassem (2015) created an Industry Foundation Class (IFC) compliant 4D tool for workspace management. The methodology and the tool provide a holistic solution to the approach of workspace management through the allocation of workspace to site activities, the detection of congestion, special and temporal conflicts and their resolution within a 4D environment in an interactive real-time manner, aided with analytic data from a centralized database.

Finally, an interesting prototype was proposed by Zhang et al. (2015). It is BIM enable approach for activity-level construction site planning in order to improve construction safety and reduce site congestion. The method includes an automated workspace visualization in BIM using an ontology approach previously formalized in (Zhang et al., II, 2015), an integration with a Global Positioning system (GPS) data loggers attached to the hardhats of work crew obtaining at least a visualization checking of workspace conflicts by using a commercial BIM platform.

The review has shown that, as regards on *generation of site workspaces* 2D/3D modelling environments, BIM-environments -both IFC-compliant and non-compliant- generate workspaces by using mark-up. Design automation in spatial modelling is still missing, and current studies on the topological interactions among workspaces and building objects as well as integrated methodologies for on-site workspaces planning are currently insufficient.

(**b**) On ***conflicts detection among workspaces*** the reviewed approaches can be grouped in three research branches that study:

1. detection of physical conflicts between the site workspaces;

2. detection of schedule conflict which means the detection of a temporal overlap between tasks that is mainly taken into account by the models which use traditional representation of the construction process (i.e., Gantt Chart and Network Diagrams);

3. site congestions identification (Zhang, 2015) that considers the ratio between the volume of resources occupying a workspace and the volume of the workspace which is available on site for a given activity or a set of activities. Often defined as 'scheduling overlapping ratio' (H.

Moon et al., 2015).

(**c**) For ***workspaces conflict resolution***, different methods have been reviewed.

They can be schematized as followings: (a) mathematical algorithms; (b) artificial intelligence methods (i.e., genetic algorithms, fuzzy logics and ant column models); (c) rule-base heuristic approaches supported by databases.

Many of these approaches reveal too complex mathematical models that lose sometimes connections with real dynamics of construction sites, can directly manage only a small part of a given building project, require an extremely knowledgeable on the proposed mathematical model which cannot be used by project managers. They almost always go unused.

2.2. Dynamic site layout planning and spatial scheduling

In this regard, even if it seems to be a more closed field, the *Site Layout Planning* aims to increase safety and productivity of a given project as well. This field has attracted many researchers in the past three decades. Several models have been developed with the common objective to determine the optimum location of site objects in the available space on site, while considering the workflows among objects. Even if all these models share the general objective, they used different approaches to define and address the problem. Leaving aside the first generation of site layout models which ignored the changes that occur within the site environment, the second generation of site layout models consider the importance of incorporating the time factor. These models are called '*Dynamic*' Site Layout Planning.

Briefly, they model site facilities setups, relocations and demolitions across the construction stages. explored the design constraints for introducing site facilities inside a construction site, changing the site facility and material demand positions and taking into account the actual use of site space over time, constraints on available locations within the site and the cost of site facility relocations (Zouein and Tommelein, 1999). They developed a mathematical model to optimize the site layout to avoid potential spatial conflicts and to optimize the relocations cost.

Elbeltadi et al. (2004) applied a genetic algorithm to solve the dynamic layout problem with safety zones, but did not consider the facility relocations. Moreover, a 4D CAD integrated site planning system that integrated schedules, 3D models, resources and site spaces for dynamic construction site planning was developed by Ma et al. (2005). Differently, Su et al. (2012) presented a geographic information system for the dynamic material site layout planning of building renovation projects that did not consider material flow.

2.3. Expert System methodologies and applications

The aim of this research survey has been to review methodologies for the development of a system architecture in order to select the most suitable methodologies for the application domain of this PhD thesis: the construction spatial planning and scheduling.

The success of any expert system (ES) relies mainly on the ability to formalize and represent the knowledge within a discipline. Often the knowledge is a collection of subjective, incomplete, ill-defined, and informal information. Indeed, a side benefit of expert system development is the formal organization of information that was previously unexpressed. ESs are codified as a branch of applied artificial intelligence (AI) and their basic idea is to simulate the action of an expert to solve a specific problem by using computer aided technologies.

By a full-scale investigation seven different categories have been reviewed as listed below and briefly presented:

(1) **Rule-based systems**. A rule-based ES contains information obtained from a human expert, and represents that information in the form of rules, such as IF–THEN. Rules are used to operate on data to inference in order to reach appropriate conclusion. These inferences are essentially a computer program that provides a methodology for reasoning about information in the rule base. They work on a data-base.

(2) **Knowledge-based systems**. They are also defined 'human-centered'. They are attempts to understand and initiate human knowledge in computer systems and many applications exist in the field of medical treatment.

(3) **Neural networks**. It is a model that emulates a biological neural network. This concept is used to implement software simulations for the massively parallel processes that involve processing elements interconnected in network architecture.

(4) **Fuzzy expert systems**. Fuzzy ESs use the method of fuzzy logic, which deals with uncertainty. This model, which uses the mathematical theory of fuzzy sets, simulates the process of normal human reasoning by allowing the computer to behave less precisely and logically than conventional computers. This approach is used because decision-making is not always black and white, true or false; but it can involve gray areas.

(5) **Case-based expert systems**. Their basic idea is to adapt solutions that were used to solve previous problems and use them to solve new problems. In such systems, descriptions of past experience of human specialists, depicted as cases, are stored in a database for later retrieval when the user encounters a new case with similar parameters. Therefore, the system searches for stored cases with problem characteristics similar to the new one, finds the closest fit, and applies the solutions of the old case to the new case.

(6) **Ontology-based expert systems**. They are used to develop a systematic analysis of knowledge within a domain of interest. Therefore, by using ontology-based modelling they

discretize the domain knowledge and formally describe a given problem. Then, by using reasoning mechanisms (rule-based) they operate on such a knowledge extracting a solution. This is possible transferring the knowledge-base in a machine-readable language.

Specifically, in the *field of construction* few expert systems exist and most of them were setted out from 1987 to 2005.

Among them and in the construction-related literature and textbooks some research subjects related to planning and scheduling can be found (Mohan, 1990). Main subjects include coding activities, sequencing activities, representing schedules and leveling resources. A number of automatic construction planners were developed based on artificial intelligence techniques, precisely. They define activities and their sequential relationships; some also estimate their durations. Those reviewed are the following:

- *GHOST* (Navinchandra et al., 1988),
- *Construction Planex* (Zozaya-Gorostiza, 1989),
- *ConsPlans* (Kano, 1990),
- *SIPEC* (Kartam and Levitt, 1990),
- *BUILDER* (Cherneff et al., 1991),
- *Know-Plan* (Morad 1991),
- *CASCH* (Echeverry, 1991),
- *HISCHED* (Shaked, 1995),
- *Case-Plan* (Dzeng and Tommelein, 1997),
- *FASTRAK-APT* (Lee et al., 1998)
- *CBRidge Planner* (Tah et al., 1999).

At first glance it emerged that an ontology-based system could have been the most effective for the issues of this PhD thesis. Therefore, a further investigation on the ontology-based modelling specifically in the field of construction project management has been carried out and later presented.

2.4. Ontology-based modelling in AEC Industry

Modeling plays a significant role in representing the domain of construction process. In the construction industry, process modeling is used more to support simulation. In looking elsewhere, ontologies can provide a powerful modelling approach. As defined by Gruber (1995), '*ontology is a formal representation of an abstracted view of a domain that describes the objects, concepts and relationships between them that holds in that domain for a stated purpose or concisely an explicit and formal specification of a conceptualization'*.

Nowadays, ontology-based modelling is central to many applications as largely explained in Motta (2000), such as medical and biological systems, information management and integration systems, electronic commerce and web services and themselves are used within the realm of artificial intelligence to capture knowledge, and create a model of the knowledge Base. It has emerged that in the recent year the development of domain ontologies in the AEC Industry has been identifies as pivotal point to develop knowledge management and integrated workflows (Zhou et al., 2016). An overview is proposed below.

Lima (2005) implemented the e-COGNOS platform testing the benefits of using semantic systems for adequate search and indexing capabilities. Secondly, the work they presented allows a systematic approach for formally documenting and updating organizational knowledge. Lastly, their work enhances the customization of functions in knowledge management systems. The e-COGNOS platform presented the first comprehensive ontology-based portal for knowledge management in the construction domain.

Another example is the ontology DOCK 1.0. It aims to develop a conceptual structure of key terms in the construction domain and their relationships and behavior. Besides representing concepts, DOCK 1.0 emphasizes the importance of context when representing construction knowledge. In addition, modality was inserted to allow users of DOCK 1.0 to generate a range of types of a particular concept (El-Diraby, 2013).

Akinci et al. (2010) envisioned that semantic CAD/GIS web services can provide away to address the lack of interoperability between CAD and GIS platform.

Benevolenskiy et al. (2012) developed a distributed multi-model-based Management Information system for simulation and decision-making on construction project. The major challenge of the system was the management of the information and model logistics as well as the interdependencies among the application models. In order to support the retrieval of information from different project phases, domains and organizations and their combination in coherent multi-models, an ontology framework was developed even if the same ontology was not structured with a computational language in order to customize the process.

A domain ontology for construction concepts in urban infrastructure products was developed by Diraby (2011). Wang and Boukamp (2011) presented a framework aiming to improve access to a company's JHA knowledge by using ontologies for structuring knowledge about activities, job steps, and hazards.

Zhong et al. (2012) developed an ontology-based semantic modeling approach of regulation constraints based on proposed CQIE ontology and construction process. They concluded that the proposed regulation-based automated construction quality compliance checking as a parallel activity to construction planning and execution can improve efficiency, reduce errors, and save human resources.

Recently, Zhang et al. (2015) investigated a new approach to organize, store and re-use construction safety knowledge. A construction safety ontology is proposed to formalize the safety management knowledge. It consists of three main domain ontology models, including Construction Product Model, Construction Process Model, and Construction Safety Model. The interaction between safety ontology and Building Information Modeling (BIM) is also explored.

Finally, in order to understand how other fields, which have high-level scheduling approaches, addressed the problem of scheduling activities and resources, Tab. 1 groups the most important reviewed studies.

Certainly, OZONE could provide an excellent framework of concepts for our research in terms of scheduling entities (Smith and Becher, 1997). The OZONE ontology represents a synthesis of extensive prior work in developing constraint-based scheduling models for a range of applications in manifesting, space and transportation logistics.

Scheduling Ontology Studies	Object	Construction Field	Other fields	Generic	Specific	Integration with other ontologies	Toolkit Integration
Scheduling Task Rajpathak et al. (2000)	Scheduling Cost Control		•	•		Time	•
OZONE Smith et al. (1997)	Logistic Scheduling		• Transportation logistics				•
Kasis-Sophina Hori et al. (1995)	Generic scheduling		• manufactory				
CommonKADS Gobin and Subramanian (2009)	Scheduling		•	•			
COMIREM Smith, et al. (2005)			Crisis-action logistics planning	•			
Job Assignment Ontology Rajpathak (2001)	Scheduling		•		•		•
Industry Foundation Classes (IFC) BuildingSmart (2004)		•			•	Building Structure	
Mephisto Lambert and Nowak (2009)			Military and national security domains		•		
OnSITEsimu *Proposed in this research*		•			•	Time Space Building	•

Tab. 1 Summary of the review and analysis of the scheduling ontologies developed in other research fields

Chapter 3 An ontology-based BIM expert system: methodology and development

From the literature review it has emerged that construction planning and scheduling methodologies and algorithms have known obstacles in their practical application mainly because of two facts: (a) the lack of interlinking between automated models, human planning, and digital building models (b) complex mathematical models that on one hand lose sometimes connections with real dynamics of construction sites and on the other hand can directly manage only a small part of a given building project.
Moreover, the complexity to hold together factors at play in simulating and scheduling construction site activities and the complexity to grinding out a detailed construction process simulation often overwhelm solving capabilities of human planners even if they have deep knowledge which could provide decisive assistance to a hypothetical integrated system architecture.

However, although the total automation is being mooted in most of domains, the aforementioned components should work in synergy, bringing problem-solving strength to the table joining forces to give expression to a unique system architecture. This is complicated by the facts that human planners do not reason about construction process plans at the level of workspaces as well as their temporal-space allocations because of the off-putting amount of time to model workspaces geometries and simulate all of them across time as well as their interaction.

In the light of these facts the proposed model which results in an Expert System, should bridge the gap between planners' experience, spatial scheduling algorithms and Building Information Models (BIMs). In this chapter its architecture is presented in four steps as depicted in Figure 3-1:

(1) Architecture of the Expert System: main blocks and related functions
(2) The underlying issues of the model development
(3) The proposed scheduling theoretical algorithm: procedure and rules
(4) Explanation of the operational framework to make the algorithm computerized

Figure 3-1 Development steps of the proposed expert system

Therefore, considering the open issues and objectives and focusing on the generation and allocation of site workspaces, detection of spatial-temporal conflicts, automated resolution of workspaces conflicts and resource optimization by using a constraint-based spatial scheduling algorithm, this study present a new approach based on an *Expert System* able to identify the shortest construction schedule.

In dealing with the complexity of constructions site analysis, this approach separates knowledge content into *Domain-Knowledge* of construction site entities and *Operational-Knowledge* of spatial planning and activities scheduling. The former represents the well-defined relationships of site items. The latter represents the reasoning processes that use the domain-knowledge to appraise the generation of the earliest construction completion sequence by taking into account workspaces availability in site.

We apply *ontology-based modelling* to structure the domain knowledge, and *decision-rules* to represent operational knowledge. Other reasoning mechanisms has been integrated mainly to solve the problem of finding and solving the spatial allocation and conflicts among workspaces.

The system is called `OnSITEsimu`.

Unlike many approaches that are considered insufficiently effective, mainly due to the lack of linking to BIM data, the model -as proposed in this research- should put the individual issues together, demonstrating how an expert system is an effective approach for managing construction schedule and how, by choosing ontology-based modelling approach within the seven reviewed categories, integration with BIM was made possible.

In this chapter before describing the main development issues, we first introduce relevant components of the model and then we present its operational architecture.

3.1. System Architecture: blocks and functions

`OnSITEsimu` is an *expert system* for computerizing a novel approach to the construction spatial scheduling (BIM-compliant) and it is designed to act as a human expert to solve the complex problem to identify the earliest construction sequence considering the temporal space allocation of resources in site (Krishnamoorthy and Rajeev, 1996). It could be considered as an intelligent workspaces planner and activities scheduler of the construction process. Its main characteristics can be expressed in terms of *blocks* it is composed of:

(1) Knowledge Base -KB-

The knowledge Base contains the domain-specific knowledge required to solve the problem. According to the goals, we consider the domain of building construction process focusing on site items, allocation and optimization of workspaces required to execute activities on site and their mutual relations as well as with the building objects.
Once, such a knowledge -construction site entities- is organized so that it will be ready for use

for the knowledge representation. It involves the preparation of a knowledge map, by using a specific *computational language*, and its own encoding in the KB. For this purpose, we propose the use of *ontologies* for modelling and the OWL language (Web Ontology Language) for computerizing –produce a script- such a KB and make it machine-readable.
Just a KB alone is not of much use if there are no facilities for manipulating the knowledge to deduce something from the KB itself. This is carried out by the *Inference Engine*.

(2) **Inference Engine -IE-**

The IE carries out the reasoning whereby the expert system reaches a solution. The system is designed to automate the generation of the *shortest schedule* given a number of input data (e.g., information on building objects, construction methods, resources and workspaces dimensions, etc.). It utilizes a set of spatial heuristics and predefined rules to schedule the activities in a chronological manner.
To do this, the KB and information provided by both the *user* and the given *BIM* have combined in order to infer new knowledge (i.e., the shortest possible total duration) by using a pre-designed rule-set.

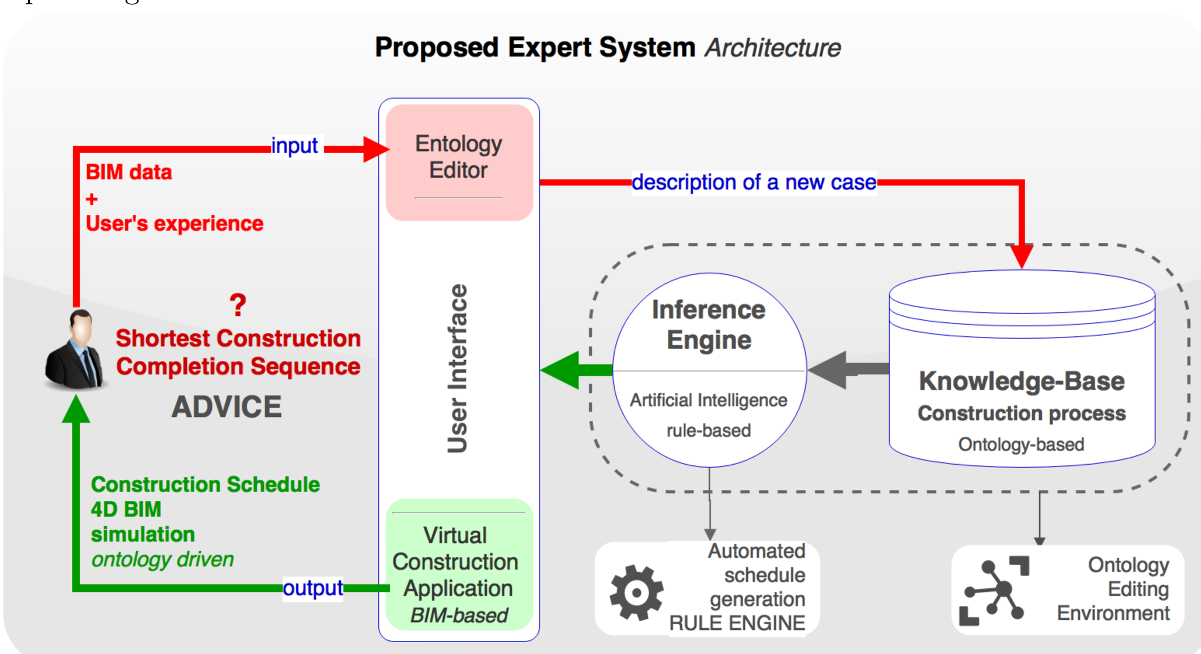


Figure 3-2 OnSITEsimu: Expert System architecture

(3) **User interface -UI-**

Generally, the user interface is the block where the user interacts with the expert system. In the proposed model, two main user interface have been identified:

- the *ontology-editing environment* that works as repository of the KB where the user adds *individuals* within the conceptual knowledge the system is equipped with. Those individuals describe construction methods he has in mind to use for each type of building object the given BIM is composed of;

- a *Virtual Construction* (VC) *application* able to simulate dynamic construction scenarios in a 4D BIM-based environment. This is crucial to visualize the result of the reasoning mechanisms (e.g., workspaces allocation and dimension as well as the activities scheduling), in terms of building production sequence with related workspaces availability.

3.2. Development issues

On the aforementioned basis, the following *development issue* have been considered:

(1) The main ***goals*** of the model are:

(1.1) To develop a unified ontological model; that means to capture the main items, also called entities, in construction site, as well as their attributes and interrelationships. It is an attempt to propose a taxonomy for construction site concepts in order to wrap the existing product model in construction, the so-called Industry Foundation Classes (IFC)[4] which doesn't contain that structure by now;

(1.2) In addition, the following categories should be considered extensively –properties- to support a fuller semantic representation of construction site activities: (a) *Building Products*, (b) *Resources and workspaces*, (c) *Time Relations* between entities, (d) *Scheduling Constraints*;

(1.3) Implement such a Knowledge-Based structure in a computer interpretable language in order to attach automated reasoning mechanisms;

Up to this point, the system thus might add pieces to a more extensive building project control in which constructability, site conditions management and productivity gain are the major objectives. By continuing:

(1.4) Introduce a theoretical spatial scheduling algorithm to find the temporal-space allocation of workspaces and define a new taxonomy to codify workspace typologies and their mutual relations;

(1.5) Reach *logical* and *technological interoperability* with a given Building Information Model (BIM) to be processed, according to the international standards of IFC;

(1.6) Automate the extraction of the shortest construction sequence.

---

[4] The **Industry Foundation Classes** (IFC) is a data model able to describe building and construction industry data. It is a platform neutral, open file format specification that is not managed by a single vendor or group of vendors. It is an *object-based file format* with a data model developed by **buildingSMART** (formerly the International Alliance for Interoperability, IAI) to facilitate *interoperability* in the architecture, engineering and construction (AEC) industry, and is a commonly used collaboration format in Building Information Modeling (BIM) based projects. The IFC model specification is open and available. It is registered by ISO and is an official International Standard **ISO 16739:2013**. (openBIM 2017)

Therefore, the scope of the listed goals can be considered to provide flexibility to construction process simulation by using a predefined structure without running manual modeling to represent and operate on site conditions.

(2) The ***role*** of the model (Figure 3-3):

(2.1) It can be seen as an intelligent blackboard that allows to reason on a solution of a construction sequence providing both (a) visualization functionalities in terms of 4D BIM simulation, model-enrichment with site workspaces, flexibility and easily updating and (b) artificial intelligence mechanisms.

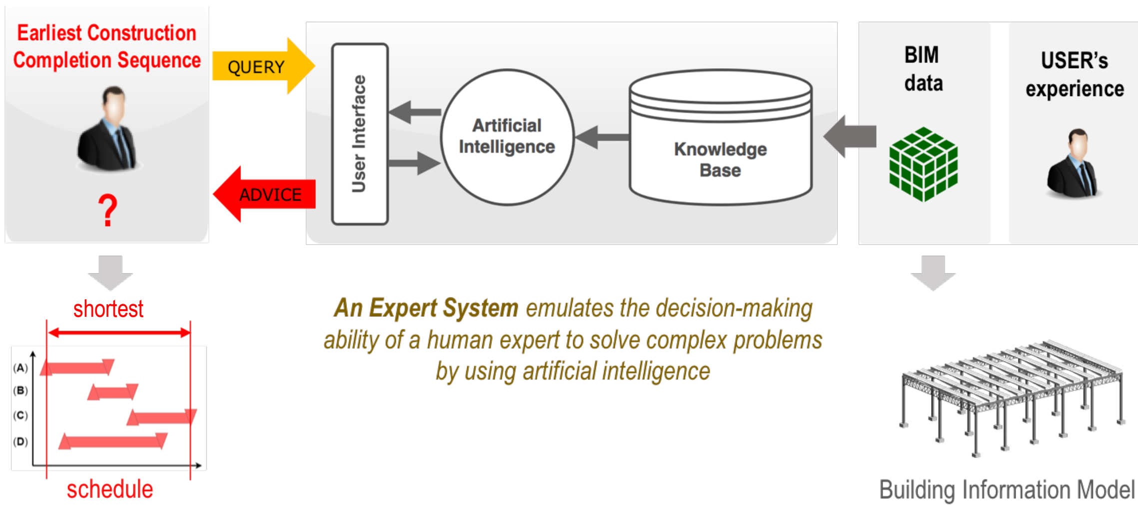


Figure 3-3 Graphical representation of the general architecture of the system and its functionality

(3) The ***problem-solving approach*** of the expert system (Figure 3-4):

In defining the problem-solving approach the following points have been considered:

(3.1) the *modeling approach* of the construction process because of its structure across time. This is carried out dividing the problem into identifiable variables at play which have drawn up the *modeling domains*. These domains are: (a) ***Building***-*Model*, (b) ***Workspaces***-*Model*, (c) ***Scheduling***-*Model* and (d) ***Time***-*Model*. Each domain has been coded by using a separate ontology which subsequently has been merged with the others generating the ***Process***-*Model*. It works as a neutral model, which maps concepts, relationships and properties of a generic construction process. This ontology will be filled in with specifications from the application domain (the given BIM) and it will compose the *Knowledge Base* (*Chapter 4*) ready to feed the Rule-Engine.

(3.2) the combination of such a knowledge base with a *workspace planning algorithm* and an *activity scheduling algorithm*. In this sense the system combines the actions of:

- an *automated planner* which guides the system when it comes to finding a solution of the optimal workspace allocation for all those construction methods the given BIM requires by using a computational algorithm described in *Chapter 9*;

- an *automated solver* which guides the scheduling strategy by using a rule-engine, again based on ontological reasoning.

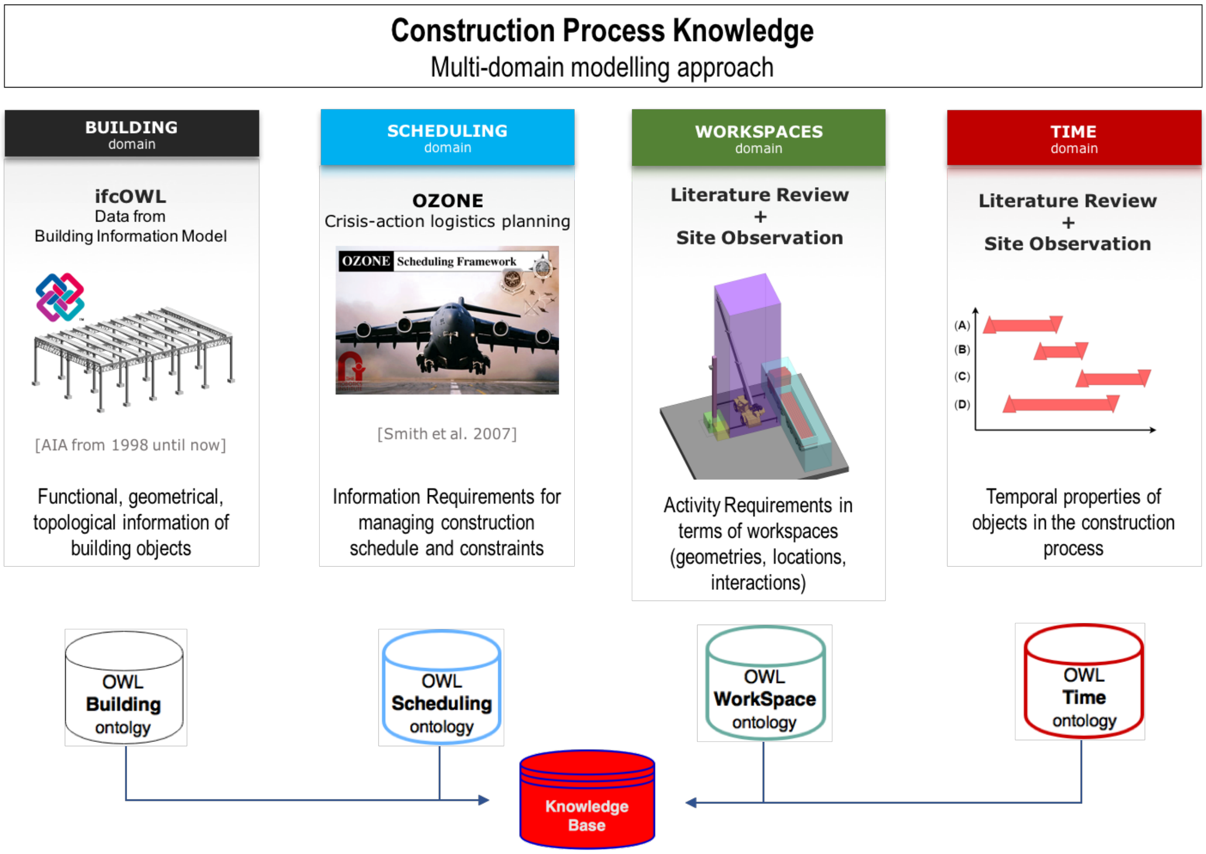


Figure 3-4 Graphical representation of the modelling approach to define the knowledge-base

3.3. Construction spatial scheduling algorithm

As mentioned before a theoretical *spatial scheduling algorithm* has been designed to drive the research in developing the system architecture and identify the construction sequence as well as start time and relationships (e.g., Finish-to-Start, Start-to-Start, overlapping, etc.) of each activity and workspace to obtain the shortest possible total duration of a given BIM. The algorithm has been created as a procedure that utilizes a set of predefined rules to schedule the activities in a chronological manner and seek the minimum required time to construct the building by extrapolating the specific construction sequence of building objects which strictly might depend on workspaces dimensions and allocations suggested by the system itself. It has been necessary to identify the steps of analysis as well as highlight those that would have required further automated reasoning mechanisms (Figure 3-5).

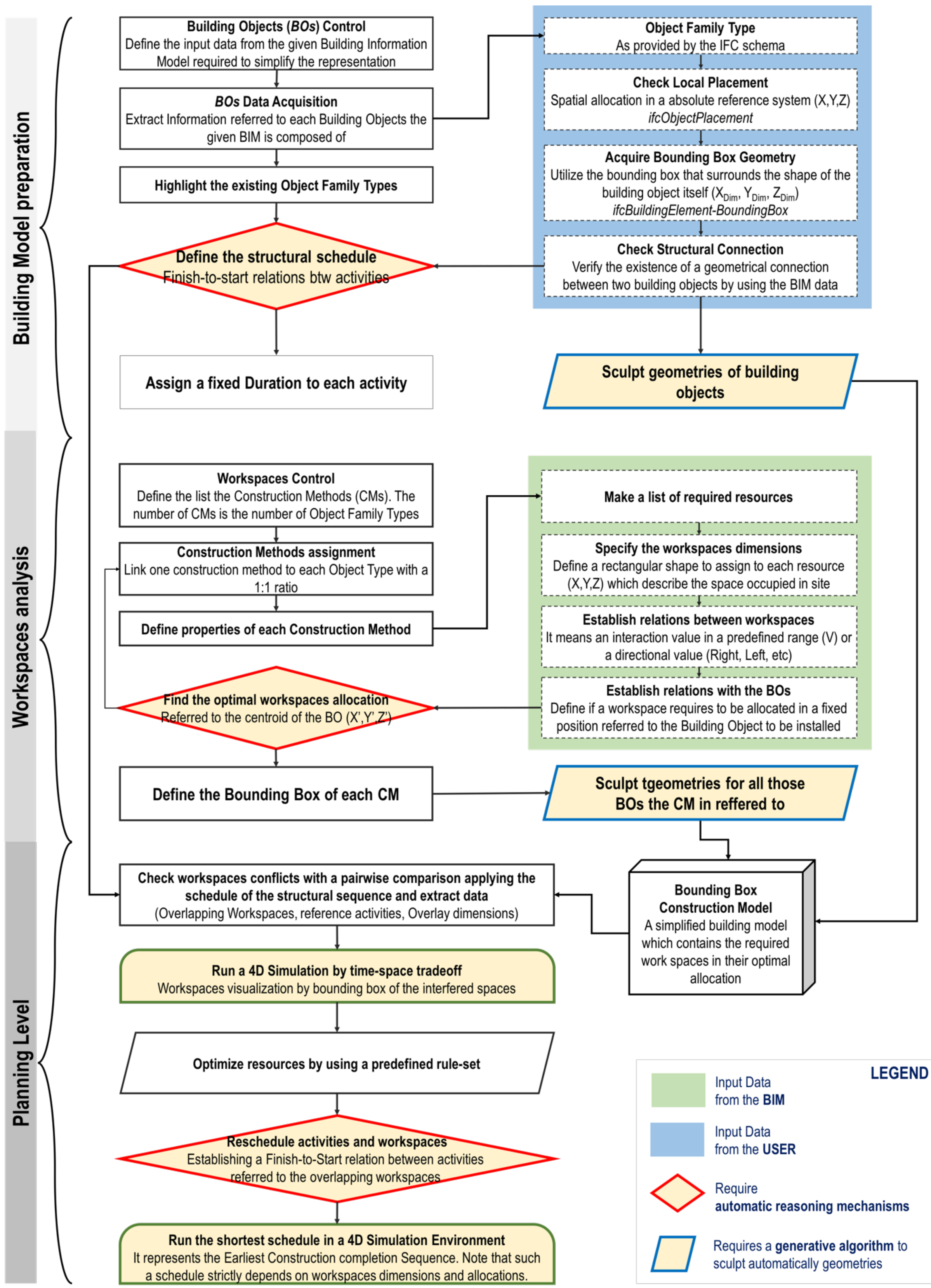


Figure 3-5 Graphical representation of the proposed spatial scheduling algorithm

3.4. Operational framework and model computerization

The spatial scheduling algorithm, as described in the previous paragraph, must be operational by ensuring a coherent data management protocol. It has been possible by using the ontology-based modelling, which has been designed with a view to operationalize this theoretical algorithm.

In the following paragraph, an overview which describes -step-by-step- the proper functioning of the designed system is presented (Figure 3-6). It integrates a number of *operation modules* that encompass the identification of problem domain, the analysis of knowledge content as well as the development of reasoning processes aiming to automate production of the shortest schedule as graphically depicted in the figure below.

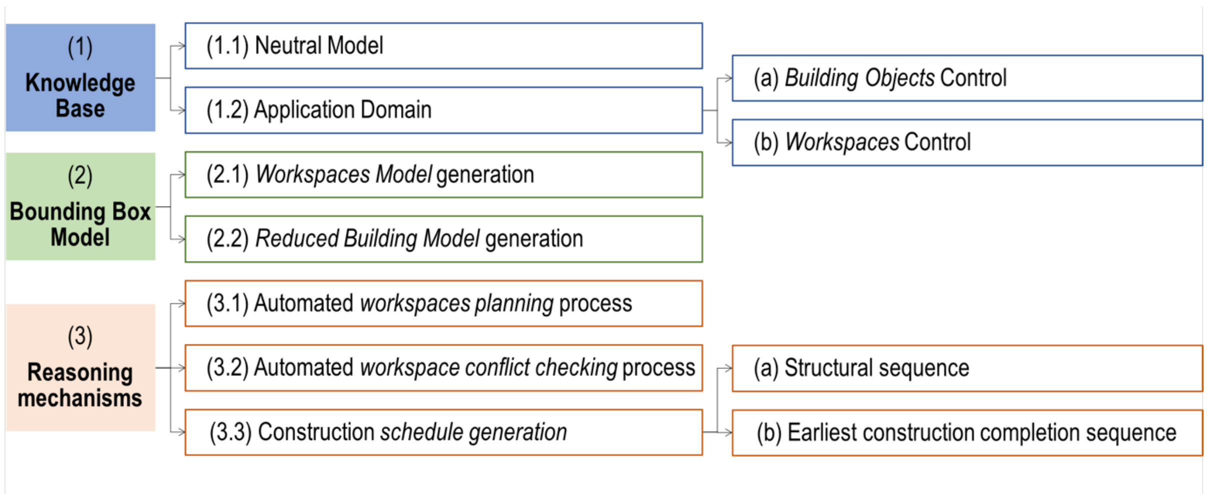


Figure 3-6 Operational modules for the proper functioning of the proposed system

1. **Knowledge Base (KB)**:

As it stated in the previous paragraph, all the necessary input data to launch the spatial scheduling algorithm are stored in a predefined Knowledge-Base in the form of *ontologies* by using classes, relationships and properties; all those traduced in *Web Ontology Language (OWL)* by using an ontology editor called *Protégé* which allows the system to define, operate and export the KB in the same format (*.OWL). Within the KB, we need to draw a distinction between the *Construction Process Ontology*[5] which contains generic entities -called *neutral model*- and 'individuals' which specify the entities for the given case study on which the system will need to operate -called *application domain*-.

Their integration composes the Knowledge-Base which works as a structured knowledge repository for the system. The reason of this specification is that the knowledge representation is uncoupled from the user who simply fills it in.

[5] For the aim of this research the terms *model* and *ontology* are interchangeable.

1.1. Neutral Model

The Neutral Model provides the knowledge sources in the form of ontologies. It is generic, such as a *conceptual data model* which contains all the possible items a construction process simulation should consider such as type of site resources, workspaces, scheduling constraints, building elements, time relationships between entities and so forth. For each site entity, a predefined *property set* has been designed. Entities, relations and properties are not randomly assigned but they come from an in-depth study as later specified in *Chapter 5,6,7,8* even because they are necessary to activate the Rule-Engine, which will be able to modify the KB on the inside by means of predefined rules as specified in *Chapter 9*. A badly structured knowledge-base would not allow the Rule-Engine to operate for generating the construction schedule.

The result is a *construction process model* that includes nodes (entities) and intelligent relationships in the form of an ontology. In Figure 3-7 just a preview of some entities, as incorporated within the system, by means of the ontology editing environment, are shown in order to provide an intelligible model explanation.

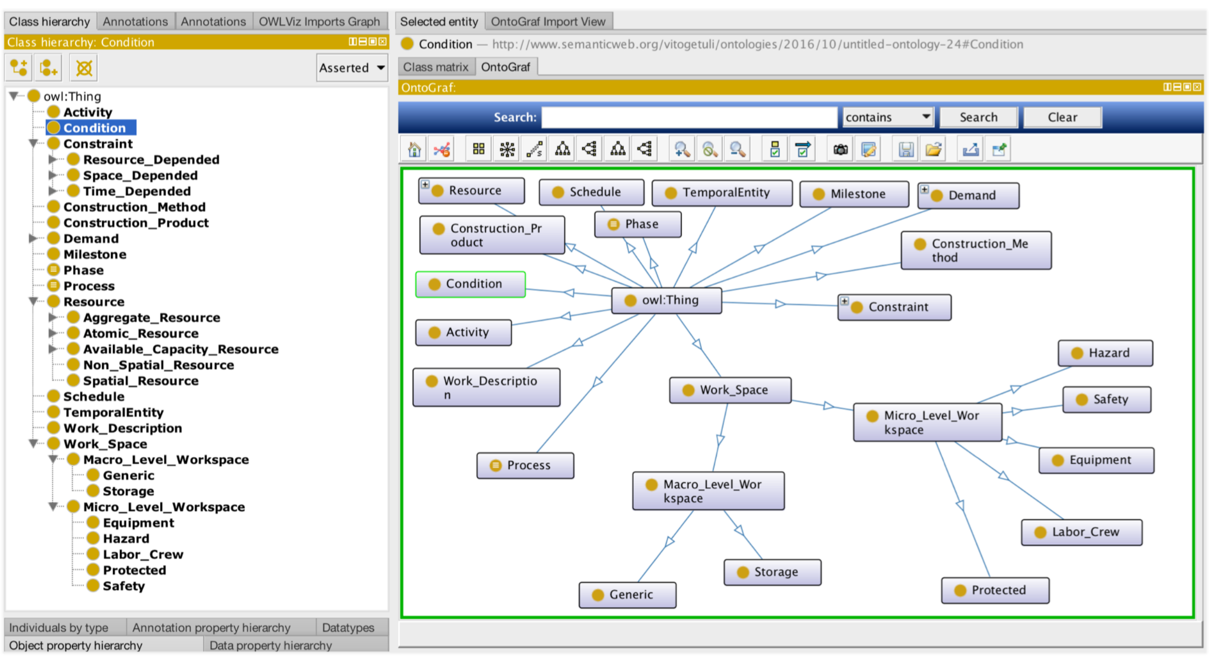


Figure 3-7 Preview of the construction process entities used in the model preparation to structure the knowledge base. Screenshot of the ontology-editing environment *Protégé* and dynamic-graph visualization of entities (right)

1.2. Application Domain

At this step, the conceptual data model (Figure 3-7) is transformed into the corresponding *physical data model*, called Application-Domain, with information originating from the specific building project (BIM) on which the system will start to operate. Here, the planning user plays a pivotal role infact he provides properties of the construction methods he intends to use specifying entities within the ontology. The transformation takes place by using two clusters of information for building project control and workspaces control:

1.2.1. Products information for the *building objects control*

Having the IFC ontology[6] at our disposal from the BIM application with which the digital Building Model was produced (*Autodesk Revit®* in this research), we are able to generate OWL-instances based on IFC-instances (Figure 3-8) by using a IFC-to-OWL conversion process as codified by *Ghent University's 'IDLab'* in (IDLab 2013).

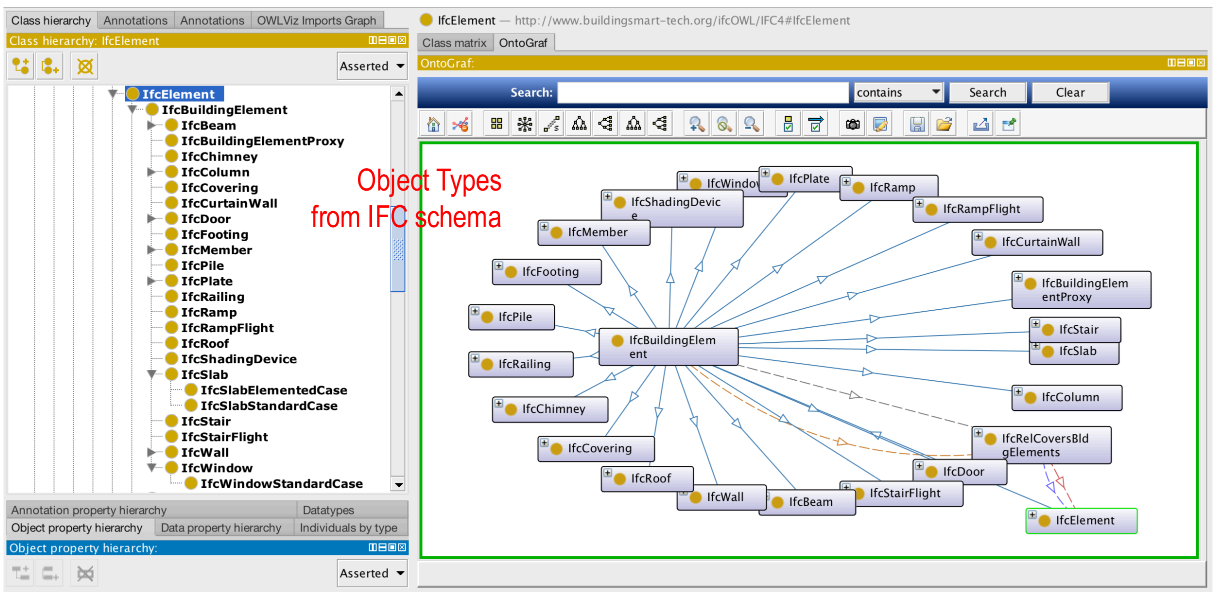


Figure 3-8 IFC building entities included in knowledge-base in order to acquire essential building information for system operation. Class-hierarchy on the left side and dynamic-graph on the right side.

By doing so the *Construction Products Ontology*, as specifically codified in *Chapter 7* for the proposed system, is filled in with building objects information, specifically selected, such as '`Local Placement`' on X, Y and Z-axis of a constrained reference system (relative to the building grid axes), '`Bounding Box Geometry`' that surrounds the shape of the building object itself (X_{DIM}, Y_{DIM}, Z_{DIM}) and '`Structural Connection`' which is the appointed entity by the IFC-schema for representing structural supports or connecting elements (nodes) of a given Building Product with others. This step is crucial to highlight building objects types (e.g., `IfcBeam`, `IfcColumn`, `IfcWall`, `IfcRoof`, `IfcRamp`, `IfcWindow`, `IfcSlab`, `IfcDoor`, etc.) and to produce a simplified building model with a bounding box representation of the building objects as well as to automatically generate the structural schedule of the given BIM by using the Rule-Engine that will operate on such information.

[6] Using an object-based inheritance hierarchy, IFC defines three abstract concepts as well as OWL Ontologies: object definitions, relationships, and property sets, whose specialized sub-classes are used to define a given BIM model.

1.2.2. Construction methods information for the *workspaces control*

The Workspaces-Ontology (*Chapter 6*), as codified in the same language (*.OWL), represents the conceptual data model which captures the workspaces properties (e.g., `Dimensions`, `Orientation`, `Movability Level`, etc.), their mutual relations (e.g., `Topological Interaction`, `Interaction Value`, etc.) and relations with the building objects for each Construction Method. All those properties have to be specified by the user. In the ontology framework, the number of construction methods equal the number of Object Types, thanks to a one-to-one association ratio. This data structure supports an automated reasoning mechanism via a built-in algorithm (*Chapter 9*), able to find the optimal workspaces configuration pattern within the site environment.

Once the Knowledge Base -Neutral model plus Application Domain- is completed, it works as a *structured data source* for the reasoning mechanisms as described below at point 3.

2. 'Bounding boxes model' generation:

2.1. Workspaces Model

Once the user defines the workspaces properties of each resource assigned to a specific construction method according to the *Workspace-Ontology*, the geometries are automatically generated at one time by means of a specific built-in algorithm, specifically developed for this research by using a visual programming environment -BIM-compliant- called *Dynamo*®. Therefore, the workspaces can be visualized in a BIM-based modelling environment *Autodesk Revit,* the same application the given BIM was produced, but only afterwards their optimal layout location will be determined (left side of Figure 3-9).

2.2. Simplified building model

Just like the previous one and using the *Dynamo*'s script, a simplified building model, later called '*reducedBIM*' which utilizes the bounding boxes that surround the shape of 3D elements for each building objects is generated.

3. **Reasoning Mechanisms**:

Once things get this far, the system requires three different automated reasoning mechanisms according to the proposed scheduling algorithm:

(a) one able to find the *optimal workspace configuration pattern* for each construction method, based on constrains defined by the user and working on information include within the workspace ontology (point 3.1);

(b) one able to check *workspaces conflicts* once that all required workspaces geometries for each building objects are sculpted via the built-in algorithm (point 3.2);

(c) one able to produce *construction schedules* on the base of a predefined rule-set, included in the Rule-Engine, that interacts with the knowledge-base by modifying relationships among the *individuals* (point 3.3).

Specifications of each reasoning mechanism as well as their computerization are given below.

3.1. Workspaces planning process

Once the workspaces properties for each construction method are defined by the user (e.g., `Dimensions`, `Interaction values`, etc.), it remains to find their optimal layout allocation (constrain-based) with reference to the building objects. This is carried out by using a configurational analysis based on *Space Syntax Methodology* (Hillier 2007) which is a calculation technique in environment that enables parametric manipulation of geometries -*Grasshopper*-. The workspaces configuration pattern is generated in the form of a planar graph by using a bubble diagram, which is deducted by Nourian et al. (2013) algorithm and especially customized and integrated for our model. Such a built-in algorithm works on *Grasshopper*, a parametric modelling tool for CAD Environment.

In this way, the expert system extracts the coordinates of workspaces allocations on the X-axis and Y-axis at the same height (Z-axis) of their connected building object but, this time, by using a relative reference system, centered on the reference object (Figure 3-9-B).

The figure below shows a preview of the graphical results of the workspaces planning process for a specific construction method –the column installation- included in the case study:

(i) on the left side, the workspaces geometries together with the building object are automatically sculpted and randomly located in a BIM-modelling environment,

(ii) then their optimal allocation is generated and finally

(iii) the workspaces a reallocated by using the new coordinates as suggested by the algorithm itself.

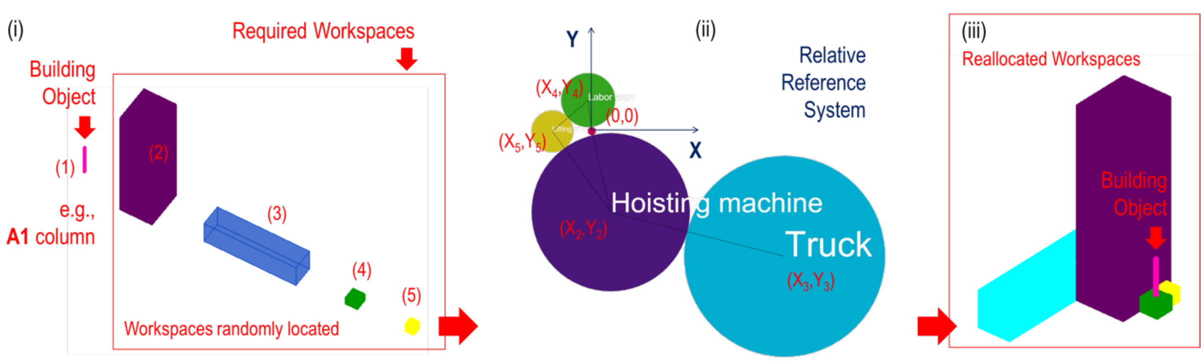


Figure 3-9 Graphical outputs of the workspaces planning process to define the spatial configuration pattern of each construction methods

3.2. Automated workspace conflict checking process and 4D simulation

At this point, the system is carrying a building model -stored in the form of ontologies and visualized within a BIM application- which includes the building objects and all the needed workspaces in their optimal layout allocation as defined by the previous planning process. In view of the axiom that two activities cannot run concurrently if their workspaces clash, the system requires a *workspace conflict detection process* (Figure 3-10).

This process is carried out by temporarily getting out of the ontologies, by means of a *time-based clash test* supported by the scheduled structural sequence as it was generated by the Rule-Engine according to (point 3.3.1.). This choice is judged to optimize the number of detected conflicts since, for example, the workspace that handles the activity of a column installation will certainly conflict with the workspace required by the activity of the beam installation structurally linked with the same column. If been checked, it would be an inconsistent conflict for the scheduling purpose.

Therefore, after completing the schedule overlapping identification, the physical conflict verification of the workspaces starts to review the conflicts. The activities that do not have overlapping schedules are excluded from the checking process because the two activities do not have the workspaces concurrently. The physical conflict amongst the workspaces is determined by an *adjacency distance* that calculates the shortest distance on the external specific surface between two workspaces. Once the workspace conflict verification process is completed a structured *clash report* is extracted in the form of strings with the (**.txt*) extension which is once again imported within the ontology in the form of workspaces properties in order to provide the system with all those required data to activate again the Rule-Engine (*Chapter 9*) able to solve, at this time, the conflicts by establishing a new time relations between all the conflicting activities by using a pairwise comparison (specification in Time-Ontology in *Chapter 8*.

The Report-Tab includes the following data for each clash result: `Item ID`, `Clash Point`, `Start Date`, `End Date`, `Distance`, `Grid Location`.
Such a conflict checking process together with the checking rules has been computerized in *Navisworks*® (4D BIM-based simulation environment) and integrated within the system by establishing a coherent information flow.

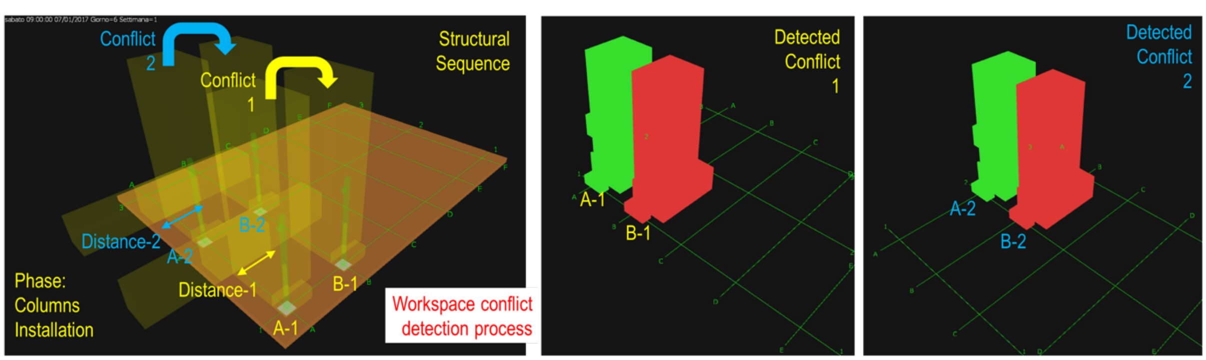


Figure 3-10 Graphical output of the workspaces conflict detection process

Figure 3-10 shows an extrapolation of detected conflicts, considering only four columns, from the case study during their installation phase.

3.3. Construction Schedule Generation

The Knowledge-Base combined with the Rule-Engine compose the core of the proposed expert system. This latter is used to apply, on the KB, a set of predefined *reasoning rules*[7] that work as artificial intelligence for the system in order to generate the structural sequence first and then the earliest construction completion sequence, also called 'shortest schedule' of the given BIM.

The Rule Engine has been implemented in the same ontology-based editing environment *Protégé*® due to the fact that, doing so, rules can automatically modify relations and properties within the knowledge base. Rules provide the description of how to solve the scheduling problem by using an IF-THEN structure that relates given information -*facts*- in the IF part to some *action*s in the THEN part. The rules are written by using the *Semantic Web Rule Language* (SWRL) for the reason that it is well integrated with the *OWL Language* used to make the knowledge base machine-readable.

From a scheduling point of view two main characteristics of the model must be clarify:

- The model considers *fixed activity duration* without 'float'[8] also know in scheduling as 'slack'. In further detail, we do not assume that the goal is to appropriately allocate the available resources but the system explores the solution of resource allocation in which activity durations only depend on the type and number of resources allocated to the activities.

- It is a *resource allocation model*, that schedule activities when there are constraints on the total amount of available resources. For example, if two column installation activities can run together because the system did not check out conflicts between their workspaces, but both require the crane availability which is however considered a `Unit-Capacity-Resource`[9] able to support one activity at a time, the Rule-Engine schedules the activities by establishing an '`IntervalBefore`' time-relation in order to execute them in two consecutive time period. An IF-THEN rule within the Rule-Engine manages this scheduling purpose (see rule 'Strategy 5' in *Chapter 9*).

The Rule-Engine acts in two different time periods, according to the scheduling algorithm, but consequential in order to generate two different construction schedule as appointed below.

---

[7] Two different approach exist to develop automatic reasoning: Rule-Based Reasoning (RBR) and Case-Based Reasoning (CBR). Rule-based systems solve problems from scratch, while case-based systems use pre-stored situations to deal with similar new instances.

[8] The *float* is a spare time in a sequence of events and is a product of the activity durations. Therefore, it is the period by which a task can be delayed, brought forward or extended without affecting the Schedule End Date.

[9] In the Construction Scheduling Ontology (*Chapter 5*) a Unit-Capacity-Resource is as sub-class of the Atomic Resource which is the smallest resource which is not divisible and can only support one activity at a time.

(a) First Schedule generation -*Structural Construction Sequence*-

On the basis of data provided in the IFC-schema, and in particular considering the information included within the entity `IfcRelConnectsStructuralMember`, which defines all needed properties describing the connection between structural members (Buildingsmart, 2014), the system operates, by means of the Rule-Engine, by establishing new time relations between building objects defining the structural sequence (e.g., building items included in the Objects Family `IfcBeam` will be related with `IfcColumn` via `IntervalBefore` time-relation, as well as `IfcWall` with `IfcWindow` and `IfcDoor`, etc.).

Once the knowledge base has been updated by means of new time relations according to the time-ontology, the schedule is visualized in *Navisworks*® that convert the scheduling data in visual data by means of a 4D simulation (Figure 3-11).

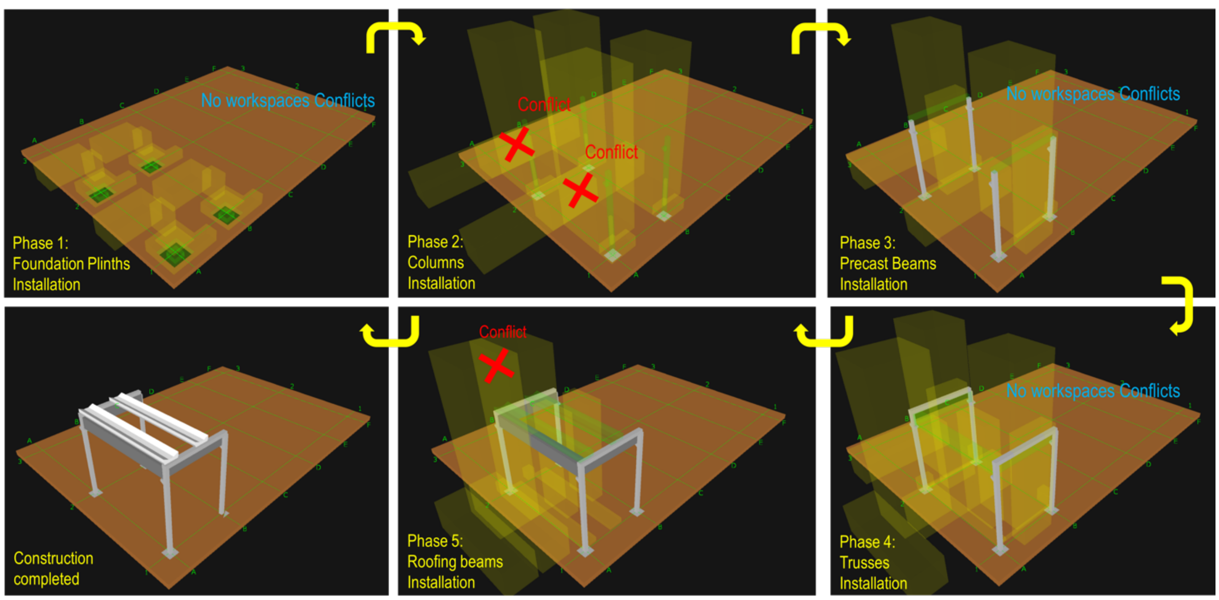


Figure 3-11 Construction schedule validation of the structural sequence visualized in 4D BIM-based simulation environment by using data provided by the knowledge base after operating the rule-engine.

(b) Latest Schedule generation -*Earliest Construction Completion Sequence*-

Having results of the 'structural schedule' as previously defined in point (a), the system proceeds by expanding such a schedule with data regarding the detected workspaces conflicts -included in the 'clash report'- (point 3.2), primarily because the conflict checking process is carried out on the structural sequence enriched with required workspaces. More specifically, in the same way as the previous schedule, each conflict is converted in a new temporal relationship among all those entities (`Building Objects`, `Resources`, `Workspaces`, etc.) somehow linked with the conflicted workspaces. Once again, this is carried out by using an IF-THEN rule implemented in the rule-based reasoning-machine (see rule 'Strategy 2' in *Chapter 9*).

This rule, combined with the others (e.g., resource levelling), updates the ontology with new relationships or properties which compose the final schedule. It represents the *Earliest Construction Completion Sequence*, the goal of the proposed expert system.

Therefore, to recap, the system architecture integrates:

(i) *Protégé*[10], as ontology-based editing environment and rule-based reasoning machine to operate on the KB according to the scheduling purposes. From the expert system perspective, which operates by using ontologies, the workflow is represented in Figure 3-13;

(ii) Then, *Navisworks*® to visualize and validate the construction sequence within a 4D BIM-simulation environment as well as to generate the 'clash report';

(iii) *Grasshopper*® as visual programming environment for the built-in algorithm able to provide the system with a workspaces configuration pattern;

(iv) finally, *Dynamo*® as visual programming environment -compliant with the BIM modelling environment- which the generative algorithm to automatically sculpt workspaces geometries has been designed.

In the following chapters, all the specifications are provided: the ontologies as well as the Rule-Engine which represent the heart of the expert system.

The operational modules before described in order to give the reader an overview of the system architecture are graphically represented in the figure below.

[10] *Protégé* ((N. Noy, 2000); (Grosso, 1999)) was developed by *Informatics Centre of Stanford University*, and it was designed to work as a platform to reduce the difficulty in knowledge acquisition, which has been recognized as a major bottleneck in developing knowledge system. It is an ontology based development, which allows users to develop knowledge taxonomy and express relationships between categories. One of its important features is the extensible architecture, which enables its integration with other applications; thus, one can easily connect external semantic modules to Protégé.

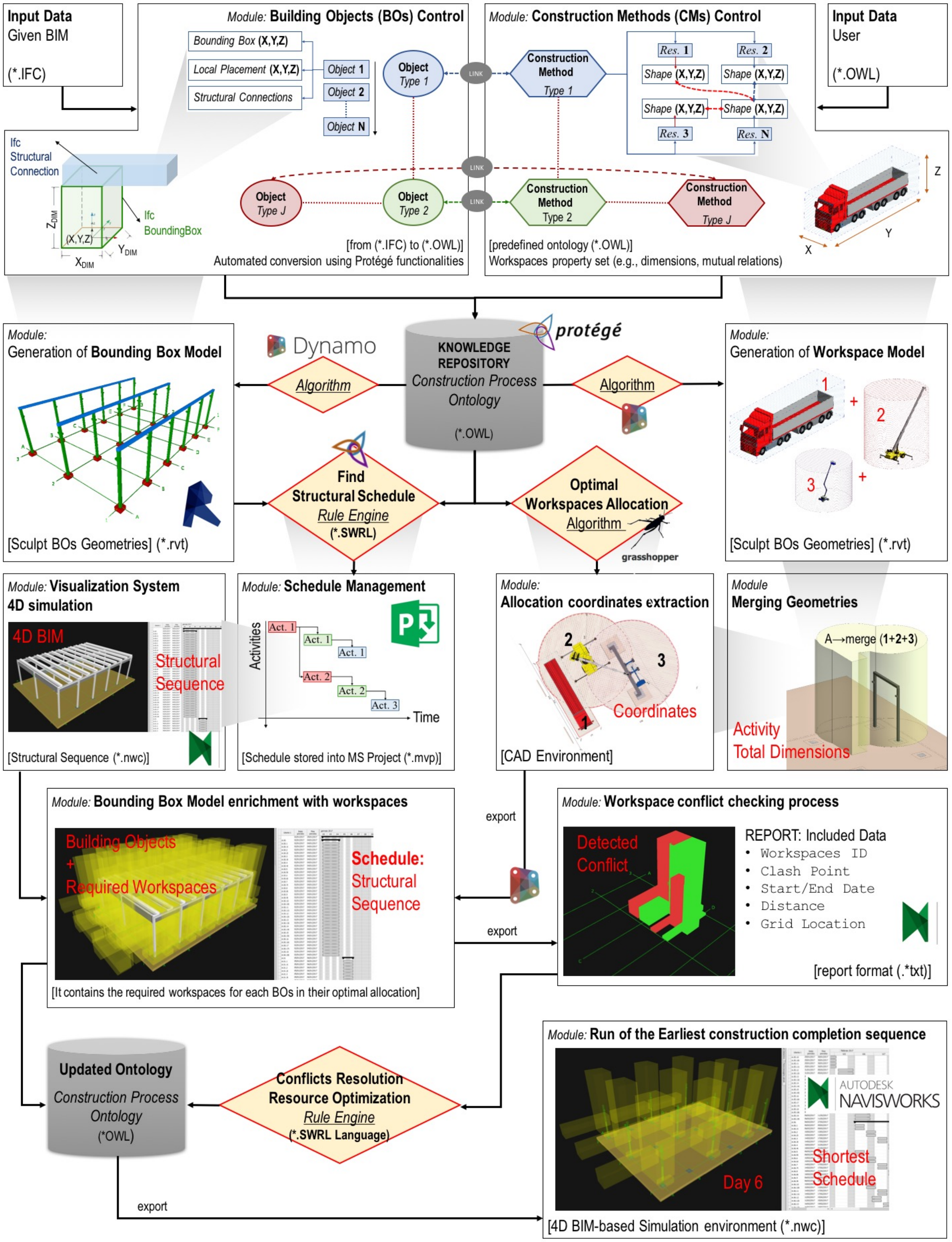


Figure 3-12 Data Management Internal Protocol: the figure shows the operational modules that guide the system to reach the generation and visualization of the earliest construction completion sequence of a given BIM

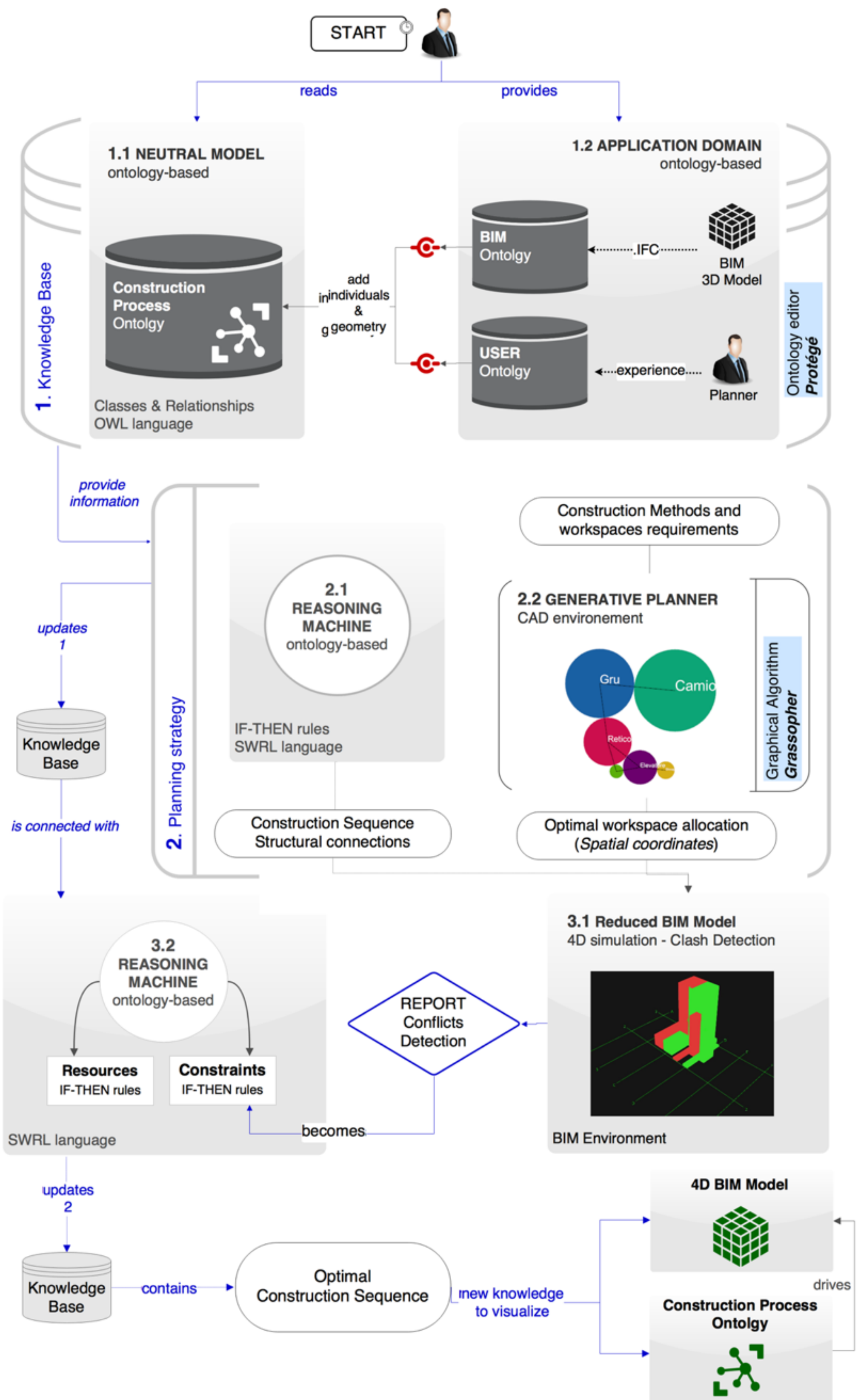


Figure 3-13 Operational Framework of the proposed Expert System from the point of view of the ontology which drives the system to operate by means of the knowledge-base and the rule-engine

Chapter 4 The Knowledge Base

As far as the chosen methodology, the preparation of information on one side of the building -provided by the given BIM- and of the construction process information on the other -provided by the user's experiences- are considered an essential part to equip the model with required data upon which to operate. This preparation process aims to define a data structure which results into a centralized knowledge-based repository (ontology-based) able to switch on the reasoning mechanisms. Such a model centric view provides robustness and flexibility to the system.

Therefore, based on the reviewed studies (Liao, 2005), a knowledge-based system supported by ontologies is proposed. By using such an approach, three components -user, BIM and reasoning mechanisms- agree with a common ontology as a specification of the shared domain of interest. The ontology is used to support communication among them, even though they may use entirely different knowledge representation mechanisms in terms of data exchange format. Once again, the objective of the Knowledge-Base (KB) structure is to formalize both the domain knowledge of construction site entities and operational knowledge of workspaces planning and activities scheduling process.

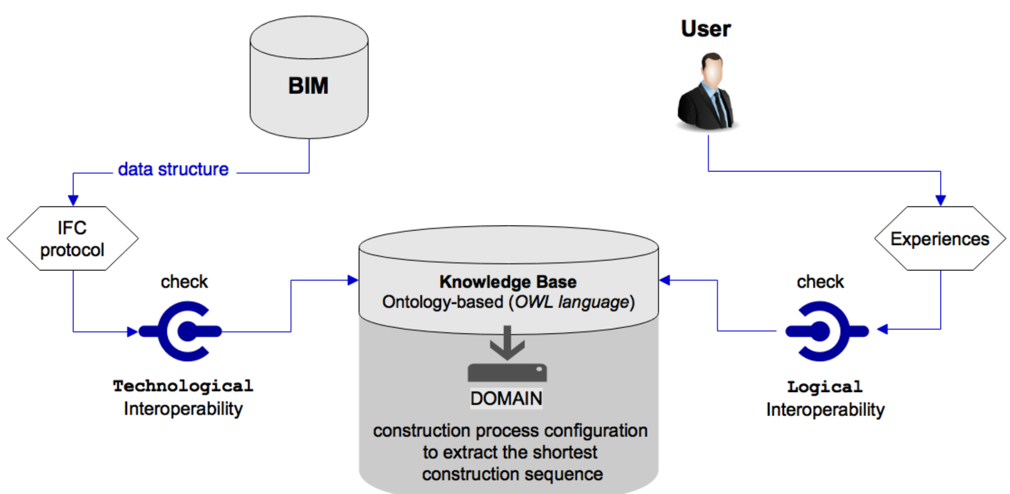


Figure 4-1 Conceptual process of knowledge acquisition in the knowledge base

The reasons, listed below, shall provide a full justification for the choice of using ontologies in order to be able to deal with a problem that other researchers try to solve with different approaches (Atkinson et al., 2007). In this regards, the comparison with *models* and *data-base* has been considered (Benevolenskiy et al., 2012).

(a) *Opportunity to consider reuse of existing ontologies.* For instance, different domain –as emerged from the literature review- need to represent the notion of time which might be include the notions of time intervals, points in time, measures of time as well as the scheduling problem which is, in the same way, an addressed question in other investigated research fields. Therefore, reusing existing ontologies is an integration opportunity, given that the system could interact with others application domain.

(b) *Ontologies support consistency checking and reasoning which is one of the object of the proposed approach.* (Models do not, database do not) One of the roles assigned to ontologies in systems engineering is to realize "intelligent databases" that can offer various kinds of reasoning services on data at runtime. Instead of the 'data integrity' used in databases, the 'consistency check' can be performed using ontologies and automated reasoning based on rule sets can be performed.

(c) *Ontology represents knowledge in an intuitive way* in the form of classes and properties (Database do not). This is an important reason if we consider that user (i.e., project managers) have to interact with them.

(d) *It is much easier to present the complex structure of the construction process by using ontologies than using a relational database.* Infact the objects of the proposed expert system ontology-based is to have a flexible body which can be easily adapted. For example, if we consider the amount of construction processes, it could be that the proposed ontology doesn't take into account some concepts or relationships or, even more, the reasoning mechanisms based on scenarios (e.g., spatial-temporal optimization of workspaces) and it can't meet the requirements expressed by the user. Using ontologies new entities or new scenarios can be added, meanwhile, in a database it can be the case that the whole table structure should be reviewed.

Going on, to make the Knowledge Base *machine-interpretable,* a set of reviewed languages exists to support the creation of ontologies (formalization of concepts, properties and relations). The most common languages that can be used for that purpose are: *KIF*[11], *F-Logic*[12], *RDF(S)* and *OWL*.

---

[11] The *Knowledge Interchange Format* (KIF) is a formal approach used for knowledge exchange among computer programs that are different in nature. The semantics of KIF are based on the correlation between the terms and sentences of the language and the conceptualization of the world in terms of objects, functions and relations. KIF uses declarative semantics for representing the meaning of expressions using first order predicate calculus and reasoning rules. This is a very early approach and lacks in its inability to transmit declarative information between large systems which is the aim of an ontology structured for the construction activity simulation.

[12] *F-Logic* is another approach, where well-defined semantics of logics are integrated with frame-based languages to provide formal semantics and resolution-based proof procedures. This was developed particularly as a database logic language comprising the object-oriented features such as object identity, complex objects, inheritance, methods, and rules.

All of them have different expressive power but have well-defined syntax, which makes them processable by computers.

In this research, we have chosen **OWL**, the *Web Ontology Language*, to compute the ontologies. It is a standard from the World Wide Web Consortium (W3C) and is the most widespread ontology language (Baader et al., 2003). The reasons of this choice are twofold and given below:

(e) As before mentioned, BIM systems and models are equipped with a standardized interface for data exchange which is the IFC (Industry Foundation Classes) standard (OpenBIM, 2016)[13]. Some pilot schemes in academic research have tried to make IFC available as an OWL ontology to allow the usage of semantic web technologies as explained in Drogemulle and Schevers (2005) and Beetz (2009).
Thanks to these research efforts, it is only a short while since the *ifcOWL* ontology, which is precisely meant to be used to allow extensions towards other structured data sets, is available. This would mean that a practical data-exchange between a given BIM and our model could be established.

(f) The possibility that the Knowledge Base can be able to rely on the ontology which underpins a BIM, would accomplish higher robustness to the expert system. That way it would also be possible to link and provide our modelling domain (classes, relationships and properties) with the logical and geometrical relationships between building objects that are contained within the BIM ontology (ifcOWL).

Based on these modelling assumptions, the developed ontological structure of the Knowledge Base is presented and specified below.

4.1. Ontological structure of the Knowledge Base

Building an OWL ontology, later called *Construction Process Ontology*, is the aim of this section. This ontology is considered to be the formal description of concepts (**OWL classes**) in charge of simulate both construction activities and scheduling process referred to them. Each concept, within the ontology, is described by using various relationships with other concepts or attributes (**OWL properties**) and restrictions on properties **(OWL restrictions**). The properties state precisely the requirements for membership of the class. Such an ontological framework combined with a set of instances (**OWL individuals)**, specifying the ontology application to a case study, constitutes entirely the **Knowledge-Base**.

[13] *buildingSMART* maintains a framework for software companies to collaborate in supporting open standards for BIM

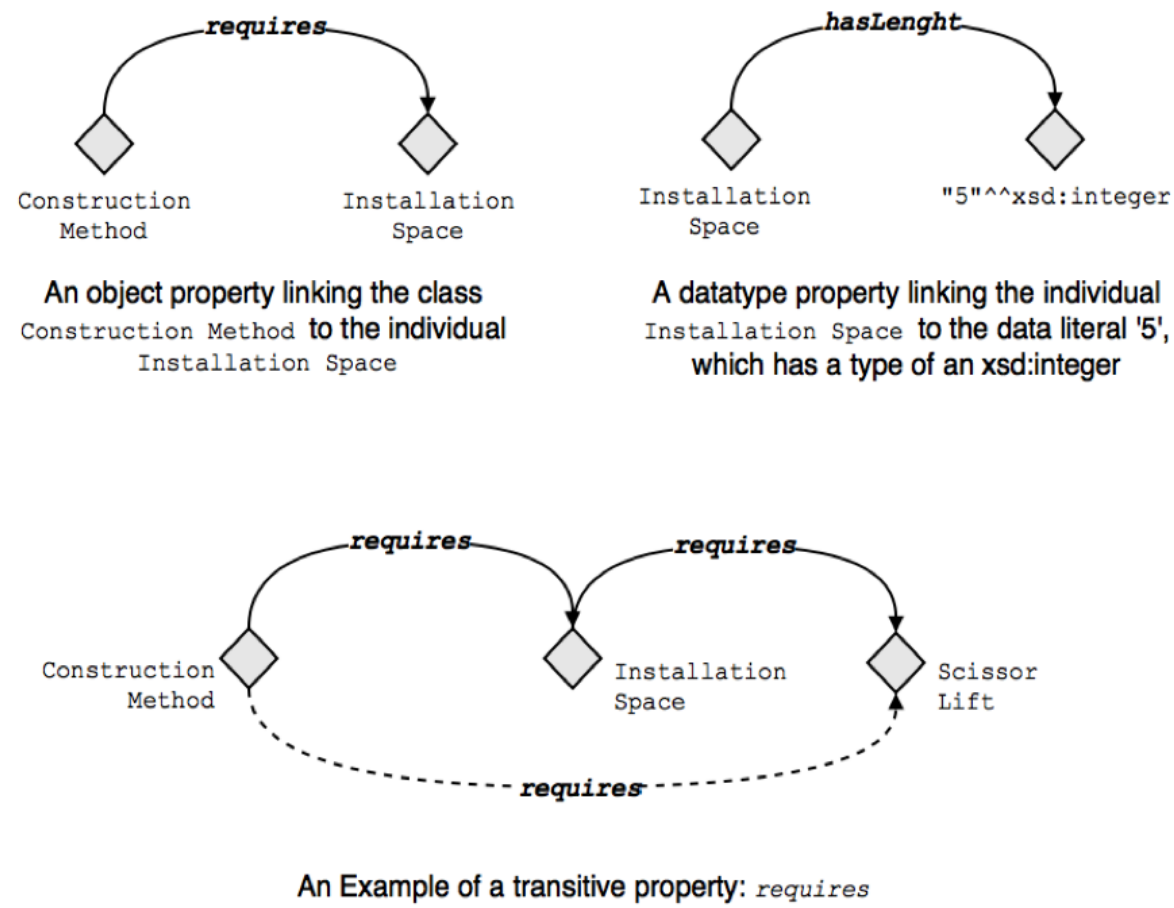


Figure 4-2 Examples of relationships (OWL properties) in the ontology modelling

More precisely 'OWL properties' are *binary relations* on classes (see '`requires`' in picture Figure 4-2) and there are two main types of properties:

- **Object-properties.** They are relationships between two classes or individuals.
- **Datatype-properties.** They link an individual to a *Datatype-value* (e.g., real number, decimal number, string, Boolean value, time instance, etc.) In other words, they are used to relate an individual to a concrete data value.

Moreover, OWL allows the meaning of properties to be enriched through the use of *property characteristics*[14] (i.e., functional -FU-, inverse -IN-, transitive -TR-, symmetric -SY-,

[14] Listed below the main typologies of object-property (relations) and their specification:
- If a property is *functional*, for a given individual, there can be at most one individual that is related to the individual via the property.
- If a property is *inverse functional* then it means that the inverse property is functional. For a given individual, there can be at most one individual related to that individual via the property.
- If a property is *transitive*, and the property relates individual **a** to individual **b**, and also individual **b** to individual **c**, then we can infer that individual **a** is related to individual **c** via property **P**.
- If a property P is *symmetric*, and the property relates individual a to individual b then individual b is also related to individual a via property P.

asymmetric reflexive -AS-, irreflexive -IR-). These textual abbreviations will be used in the ontology specification.

It is evident that classes are the cornerstone of the ontology. For example, a class of '`safety_spaces`' could represent all the workspaces with the same end-use likely to be in a construction site. Specific spaces on the application domain will be instances of this class. Classes may be organised into a *superclass-subclass* hierarchy. Sub-classes specialise ('are subsumed by') their super-classes. For example, the class of '`workspaces`' could be divided into hazard and non-hazard spaces or into paths, warehouses, material staging areas, laydown area and so forth. One consideration jumps out: the fact that the ontology strictly depends to the objectives.

In this regard, the ontology visualization can help by assisting in the development, exploration and verification of themselves. Although several visualizations for ontologies have been developed in the last couple of years. They work in the same ontology editor environment *Protégé*. After a survey[15], the Visual Notation for OWL Ontologies –VOWL- has been chosen to represent ontologies in this research (Lohmann et al., 2014). The representations are based on graphical primitives and color scheme; a selection is shown in Figure 4-3.

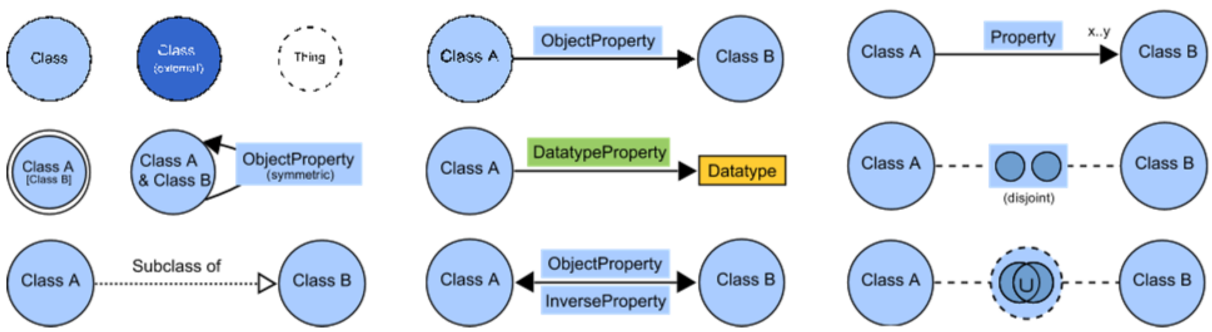


Figure 4-3 Selection of visual notions to represent ontologies after developing a specific script in OWL language (Lohmann, 2014)

With regard to development of our ontology, there is no standardized methodology (McGuinness and Noy, 2001).

The review of the ontology building methodologies that have been used in the construction

---

- If a property P is *asymmetric*, and the property relates individual a to individual b then individual b cannot be related to individual a via property P.
- A property P is said to be *reflexive* when the property must relate individual **a** to itself.
- If a property P is *irreflexive*, it can be described as a property that relates an individual **a** to individual **b,** where individual **a** and individual b are not the same.

[15] Several visualizations for ontologies have been developed. Well-done reviews are proposed in M. Dudáš (2014), Katifori (2007), Lanzenberger (2010). Comparisons of selected visualizations are given in Fu (2013) and Chan (2011). Several of the visualizations have been implemented as standalone applications, but most are realized as plugins for ontology editors like Protégé, the same ontology editor used in this PhD research.

industry are set out in the annex to these note[16].

Therefore, in this PhD thesis, the steps for developing the ontologies and corresponding deliverables are explained below and schematically depicted in Figure 4-4:

Step 1) *Investigation of the knowledge resources.* This step focuses on reviewing already existing ontologies, taxonomies, or other sources within the construction domain, and how to reuse them. In other words, ontological and non-ontological resources have been investigated. Bringing forward, clear example of a non-ontological resources was the case of scheduling and workspace ontologies in which the construction site observation proved essential to model the construction process.

Step 2) *Specification of objectives*: In order to figure out which and how many classes, relations and properties comprises the ontology, in this step the modeling objectives has been gained by answering a list of competency questions such as the following. Why is the ontology being built? What type of information should be involved in the ontology? In order to give a clearly visible structure of objectives, a graphical representation is proposed for each sub-ontology.

Step 3) *Definition of the overall framework of the ontology*: in this step a list of main selected concepts (classes) and their formal explanation are presented.

Step 4) *Definition of the topological relations and integration with other domain*: the core of the ontologies is here presented. Classes and class hierarchies are well-defined, relationships between classes are established and properties and attributes are identified according to the objectives. This is crucial infact deciding whether a particular concept is a class in the ontology or an individual instance depends on what the potential applications of the ontology are.

Step 5) *Ontology specifications and computation in the ontology editor editing environment*: the ontologies are first modelled and then rendered in a 'script' in form of OWL language by using *Protégé* (Horridge, 2011). In order to design a correct and not redundant ontology, the consistent of the ontology has been checked using an automated consistency checker.

16

Methodology	Main Steps
NeOn 2012	Specification, conceptualization, formalization, implementation, integration, mapping, and merging
SKEM (McGuinness, 2001)	Determine the domain and scope, consider existing ontology, enumerate important terms, define the classes, define the slots, define the facets, and create instances
METHONTOLOGY (M. F. Lopez, 1997)	Specification, knowledge acquisition, conceptualization, integration, implementation, evaluation, and documentation
Unshold and Gruninger 1996	Identifying purpose and scope, building ontology, integrating existing ontology, evaluating the ontology, and documentation

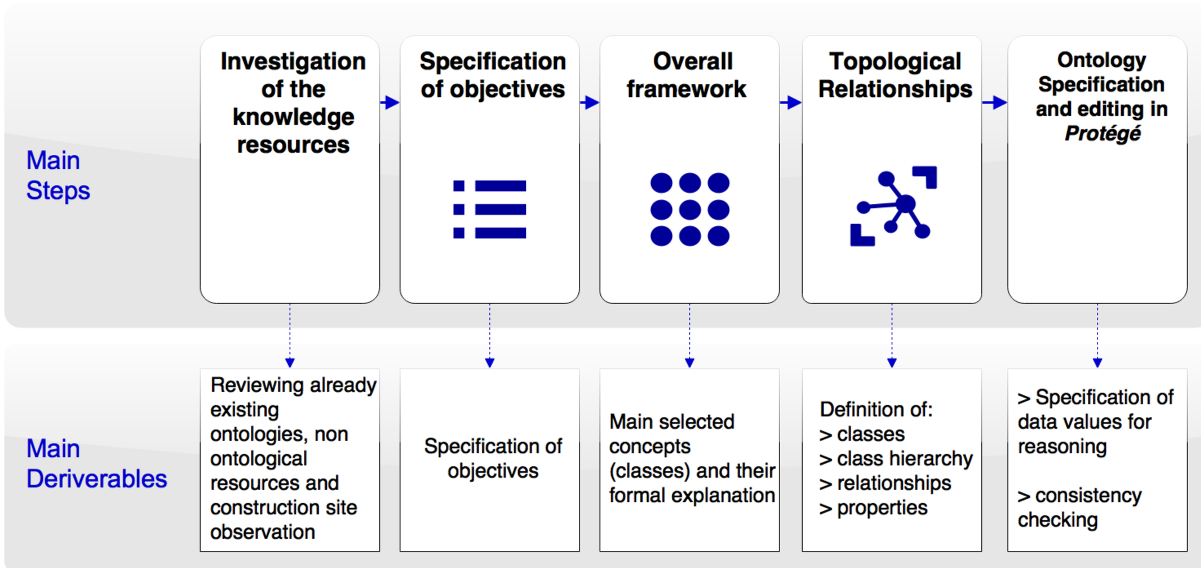


Figure 4-4 Research tasks in ontology development

4.2. Modelling domains in the Construction Process Ontology

The problem of modelling the construction process for workspaces planning and activities scheduling is the result of a complex process involving many decision variables, here defined as *modelling domains*[17]. As a first step for developing our Knowledge Base, it is necessary to define the different variables at play from which planning and scheduling approaches will be dependent.

These domains shall be drawn from the aims of our expert system that may be summarized as follow: '*the knowledge base should ensure the extraction of the shortest construction schedule that optimize workspaces allocations and solve overlapping problems of construction activities'*. It jumps out that the domain of site workspaces and their connection with building components contained in a given BIM plays a pivotal role.

That is why the proposed Knowledge Base does not follow an *all-in-one modelling approach* but analyze the individual models by considering singularities of each domains separately. This choice of a *multi-ontologies*, for modelling the construction process domain, is justified on the grounds that their interrelations can be specified in order to give a higher flexibility to the knowledge base that might be opened to future extensions in terms of others domains (e.g., risk analysis, health and safety management, paths planning, monitoring systems, etc.).

[17] In this sense, (Heijst et al., 1997) distinguish different types of ontologies, e.g. *domain ontologies*, which express conceptualizations for particular domains; and *generic ontologies*, which concepts that are considered to be generic across many domains. In *OnSITEsimu* the domain ontologies are used to model the domains of scheduling, workspaces and building elements differently form the time domain which is modelled by using a generic ontology.

Overall, within the designed Knowledge Base, four modelling domains has been identified. They are coded by using four sub-ontologies, listed below, as the conceptualization in Figure 4-5 explains.

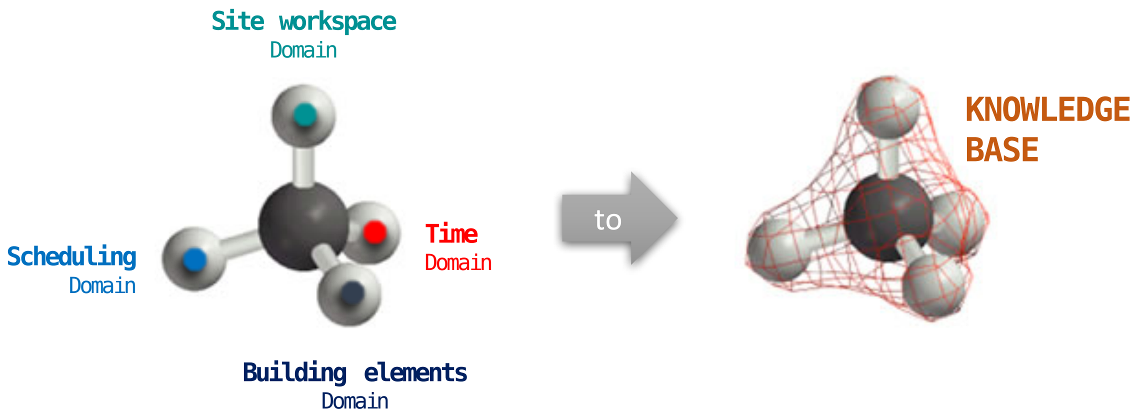


Figure 4-5 Conceptualization of modelling domains which compose the Knowledge Base

(1) *Construction* ***Scheduling*** *Ontology*: this sub-ontology contains all those elements for representing the scheduling problems and constraints. It provides a structural foundation for analyzing the information requirements of a construction schedule which should be depend on availability and typology of resources, on space-temporal constraints, on allocation of workspaces and so forth (details in *Chapter 5*).

(2) *Construction* ***Workspace*** *Ontology*: it contains the site workspaces representation and the property set able to activate the reasoning mechanisms and the built-in algorithm to allocate workspaces themselves. Infact, workspaces need to be represented with their basic geometrical and capacity properties and need to be linked to the building objects (details in *Chapter 6*).

(3) *Construction* ***Time*** *Ontology*: it is the ontology of temporal concepts for describing temporal properties of site entities in their evolution across time. It also objects to describe possible relations between time periods in order to define the temporal positions among activities, workspaces and building objects. It plays a pivotal role in developing rule-based reasoning mechanisms for minimize overlapping activities in terms of workspaces. It works as a means of connection with a Calendar to the Knowledge base with the 4D BIM simulation environment which require a construction schedule contextualized in a define time-period (details in *Chapter 8*).

(4) *Construction* ***Product*** *Ontology*: This sub-ontology represents the domain of Building Information Models (BIMs) and it describes the functional, geometrical and topological information of the building objects –products- that the Knowledge Base needs to get in order to activate reasoning mechanisms and generate the 'Bounding boxes Model'.

Based on IFC-schema and more specifically on *ifcOWL* ontology, a new sub-ontology has been specified to represent the building elements with a low Level of Detail (LoD) (details in *Chapter 7*).

The *ontological framework* is depicted in Figure 4-6 and the *configuration process* explained in the next chapters.

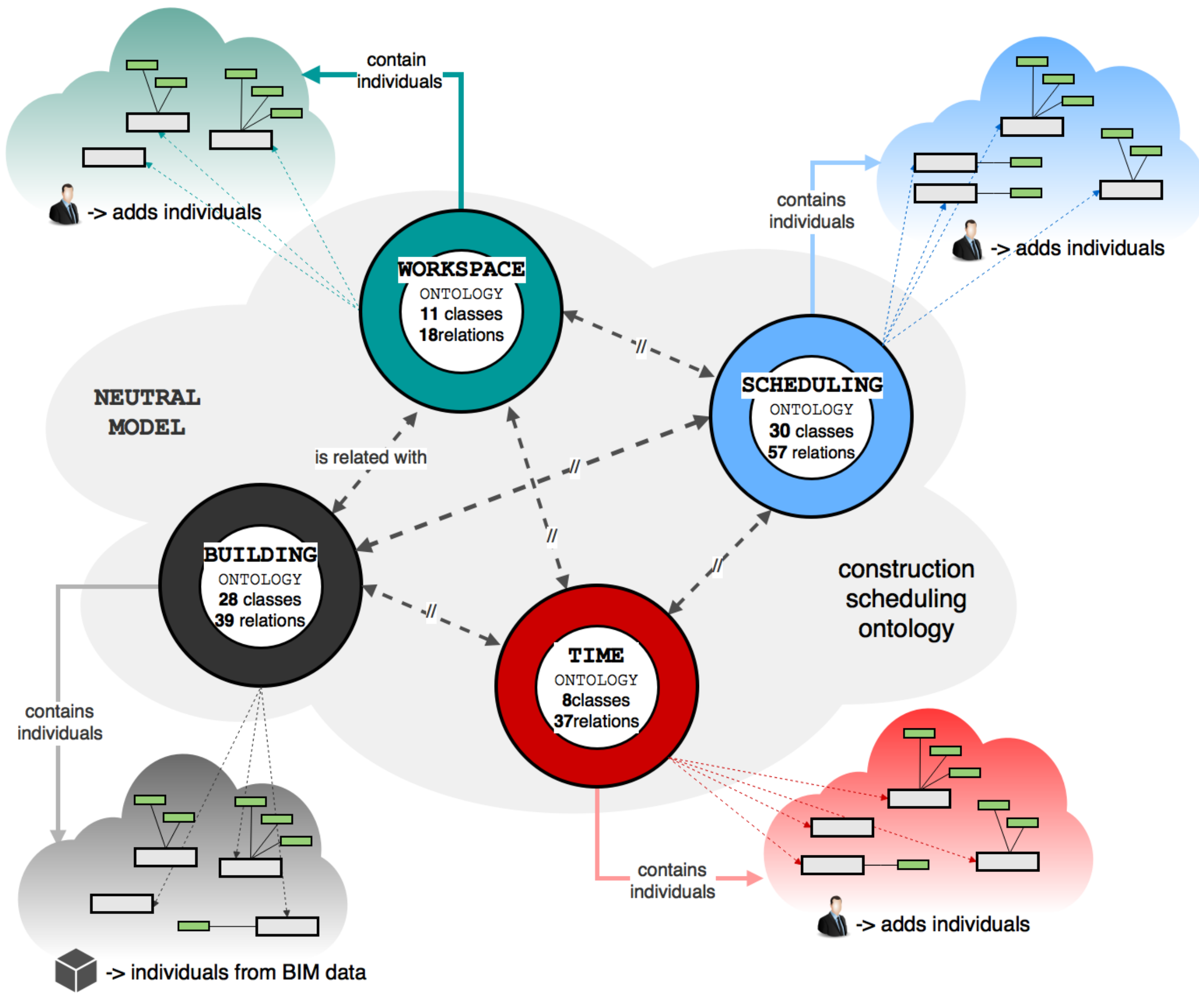


Figure 4-6 Graphical representation of the ontological framework of the Knowledge Base capturing the integration of four sub-ontologies and their specification by means of 'individuals'

In next chapters an in-depth study that has led to structure the ontologies is presented.

Chapter 5 Construction Scheduling Ontology

Scheduling is the key function contained in our system architecture. This paragraph is about details of the Scheduling Ontology which has been developed to automate the generation of a constrained-based construction schedule.

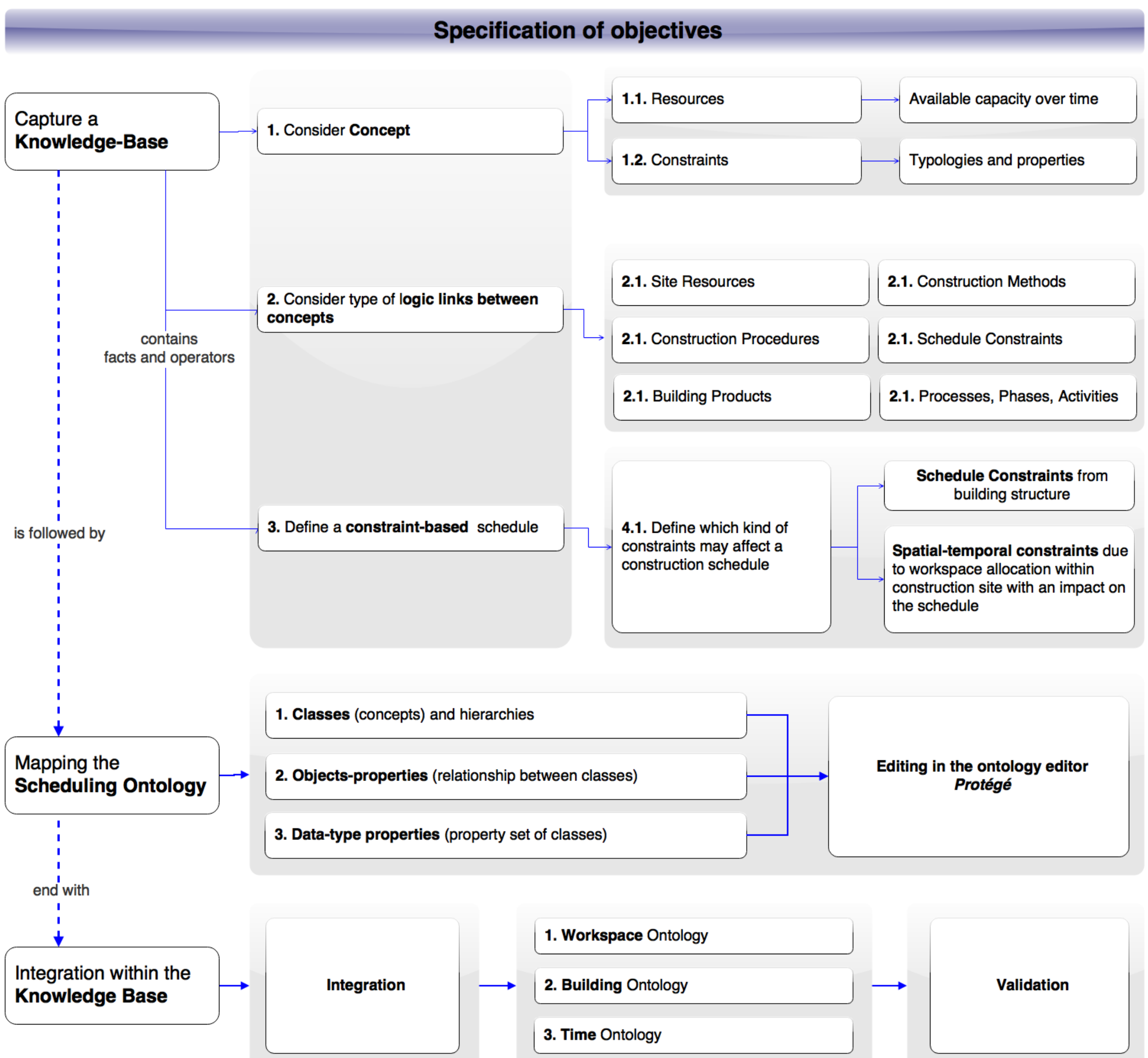


Figure 5-1 Development plan and objectives specification of the construction scheduling ontology

5.1. Specification of modelling objectives

Central object of the scheduling ontology is to define a reusable and extensible base of concepts and relations for describing and representing scheduling problems in the domain of building construction. It provides a model for describing those aspects of the scheduling domain that are relevant to identify the 'shortest construction sequence' when synchronized with the rule-engine (as presented in *Chapter 9*). This connection is established by including an operational semantics to the OWL objects.

The modeling objects are specified by means of a list, later explained in a diagrammatic form in Figure 5-1:

(i) include scheduler capabilities and experiences in the scheduling domain;

(ii) automate the scheduling process of the structural construction sequence which are user-independent;

(iii) provide a constraint-based solution framework of scheduling architecture which encapsulate reusable concepts and intelligent relations for configuring and customizing the constrain-based scheduling solution:

(iv) maximize productivity by minimizing idle time and overlapping as many activities as possible if their workspaces do not overlap;

(v) link the scheduling ontology with the building structure as contained in the Industry IFC-schema, an object-oriented format which provides a universal base for data exchange.

5.2. Overall framework of the Scheduling Ontology

In the proposed scheduling framework, the ontology can be formally represented as a mapping from a *twelve-dimensional* space. Such an input parameters provide the necessary components to specify the scheduling task, which are connected by using binary relations which specific 'property characteristics' (Figure 5-2).

1. `Construction Method`, (CM) = $\{cm_1, ..., cm_n\}$. This class is an abstract entity which describes the construction work execution. This entity drives the ontology. The construction schedule, linked to a given Building Information Model, should have Construction Methods as much as the number of Object types.

2. `Work Description`, (WD) = $\{wd_1...., wd_n\}$. It describes the construction execution referred to a given Construction Method, its spaces and resources on site by using generic terms.

3. `Demand`, (De) = $\{de_1, ..., de_n\}$. This class contains both construction procedures and safety

rules that are formally and graphically simulated by using the ‘workspace ontology’.

4. **Construction Product**, (CP) = {cp_1, ..., cp_n}. This class comprises all the building objects that are primarily part of the construction of the building itself. Its categorization comes directly from the IFC-schema as later described in *Chapter 7*. Hence, the *ifcBuildingObjects*, contained in a given BIM, are converted in individuals that are referred to as being instances of the class ‘Construction Product’.

5. **Condition**, (Cn) = {co_1, ..., co_n}. This abstract entity describes condition that must be achieved at the beginning (pre-condition) or ending (post-condition) of a Construction Method. A Condition can be expressed in terms of activities or milestone in a time period.

6. **Resource**, (Re) = {re_1, ..., re_n}. To define a Construction Method, it is necessary to choose specific Resources with a specific proposed-set. Semantics and properties of those resources vary according to the type of Resource and define their available capacity across time. A number of resources have been proposed which should be cover those required by a construction process.

7. **Constraint**, (Cs) = {co_1, ..., co_n}. Getting the Expert System to work on the solution to the given scheduling problem, constraints determination and satisfaction is essential. Generally, a constraint restricts the set of values that can be assigned to a given variable according to Smith et al. (2005). Our scheduling domain provides the means to model three types of constraints that restrict the assignment of Start and End-Times and the physical allocation in site of Resources and Workspaces related to each construction activity:

    a) **Resource-depended**. It designates the condition under which a Resource (e.g., scaffolding, labor crew, etcetera) can be assigned to a given construction activity or restrict the physical capabilities of resources to handle more activities simultaneously;

    b) **Time-depended**. It defines the possible relations between objects within the construction process (e.g., *before*, *meets*, *overlaps*, *during*, *equals*, etc.) and their time periods;

    c) **Space-depended**. It consists in a family of three sub-constraints which are strictly connected to the workspace simulation (e.g., equipment space, labor crew space, hazard space, etc.) and all those constraints which can be automatically extracted by the IFC Building Structure (e.g., if workspaces of two activities overlap, they can’t run simultaneously in construction site).

Moreover, a further classification of constraints has been introduced:

d) **User-setted**. They derive directly from the user specifications depending on his experience who directly could add constraints;

e) System-setted. Those ones that are automatically generated by the ontological structure due to properties assigned to the relationships;

f) Building-setted. They derive directly from the BIM by using transformation rules (e.g., a beam should be constructed after connected columns);

g) Simulation-setted. Those ones that derive from the 'workspaces conflicts checking process'.

8. Phase, (Ph) = {ph_1, ..., ph_n}. A group of strongly-related construction processes defines a Phase which ends with a Milestone.

9. Process, (Pr) = {pr_1, ..., pr_n}. A process represents the most abstract class that groups various activities.

10. Activity, (Ac) = {ac_1, ..., ac_n}. In the proposed architecture, a schedule is represented as a network of Activities that will produce a number of Construction Products by using workspaces. To schedule an Activity, it is necessary to choose Resources that produce the time intervals to assign to each activity depending on their capacity level.

11. Milestone, (Mi) = {mi_1, ..., mi_n}. A Milestone represents a Phase finalization connected to a given Time Instant.

5.3. Topological structure

Here, it is depicted the framework models of the scheduling ontology in terms of the main classes, properties and relations diagrams. To enhance a better explication, in body of the text that follows, a different font has been used for ontological objects and the font color is used to distinguish the belonging to one ontology that composes the Knowledge Base (Figure 5-2).

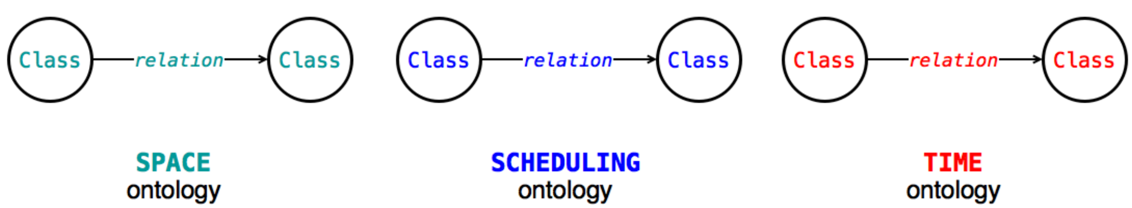


Figure 5-2 Textual notations used in the body of the text in order to distinguish the ontology objects

Core class of this ontology is ConstructionMethod. This is due to the fact that other classes depend on it. Infact by using relationships and properties, listed below, each construction method is described in terms of required resources, activities and workspaces. All these classes are inextricably linked in an intelligent framework.

Therefore, a `ConstrucitonMethod` *produces* or *consumes* a number of Construction Products. The class `ConstructionProduct` contains a list of individuals which represent the building elements and their information requirements, such as columns, beams, slabs and walls, and provide the main interface for connecting the scheduling problem to the given BIM. It follows the structure of the IFC schema and mainly includes sub-types of *IfcBuildingElement*. This building elements include major functional parts of a building.

The binary relationship between Construction Methods and Construction Products is chosen by the user. A Construction Method presumes some `Condition` could be existing before (*precondition*) or after (*postcondition*) the given Construction Method runs within the construction site. A Construction Method *isDescribedBy* a `WorkDescription` which specifies the construction execution describing allocation of spaces and required resources by using generic terms. A WorkDescription is regulated by a procedural guideline which is specified in the class `Demand` by using a set of principles or conditions which can define a `procedure` or a `SafetyRule`. This means that if the user links two workspaces to a construction method the system automatically classifies this relation as a procedure of the Construction Method or as a safety rule if a safety or hazard space is included. Each procedure or safety rule contained in a Demand *requires* a number of `Resource`.

The class `Resource` is also central to the definition of our scheduling ontology. It represents an entity which is assigned to a Construction Method for its execution. Each Resource can *handle* one or more activities simultaneously and is provided by a specific property set. These properties are all those which effect its availability and utilization in function of its specific Capacity (e.g., *hasCapacityLevel*). Making efficient use of Resources, in supporting activities, becomes the one crux of the scheduling problem which is managed by the rule-engine. A resources class hierarchy has been proposed which models each sub-class in terms of its dynamically changing amount of *CapacityLevel*. The class hierarchy is explained in Figure 5-3 and the main class restrictions which are modelled are depicted in Figure 5-4.

Going on, an `Activity` *isFollowedBy* an interrelated set of sub-activities. To define a schedule, each Activity requires `MicroLevelWorkspaces` to being performed in site. A `Process` is modelled as an abstract entity which *isComposedOf* a number of Activities. More Processes make up a `Phase` which ends with a `Milestone` and requires `MacroLevelWorkspaces` to being performed within the construction site. These relationships must follow a number of restrictions as presented in Figure 5-4 and Figure 5-5. Finally, a Milestone involves one or more `ConstructionProduct`.

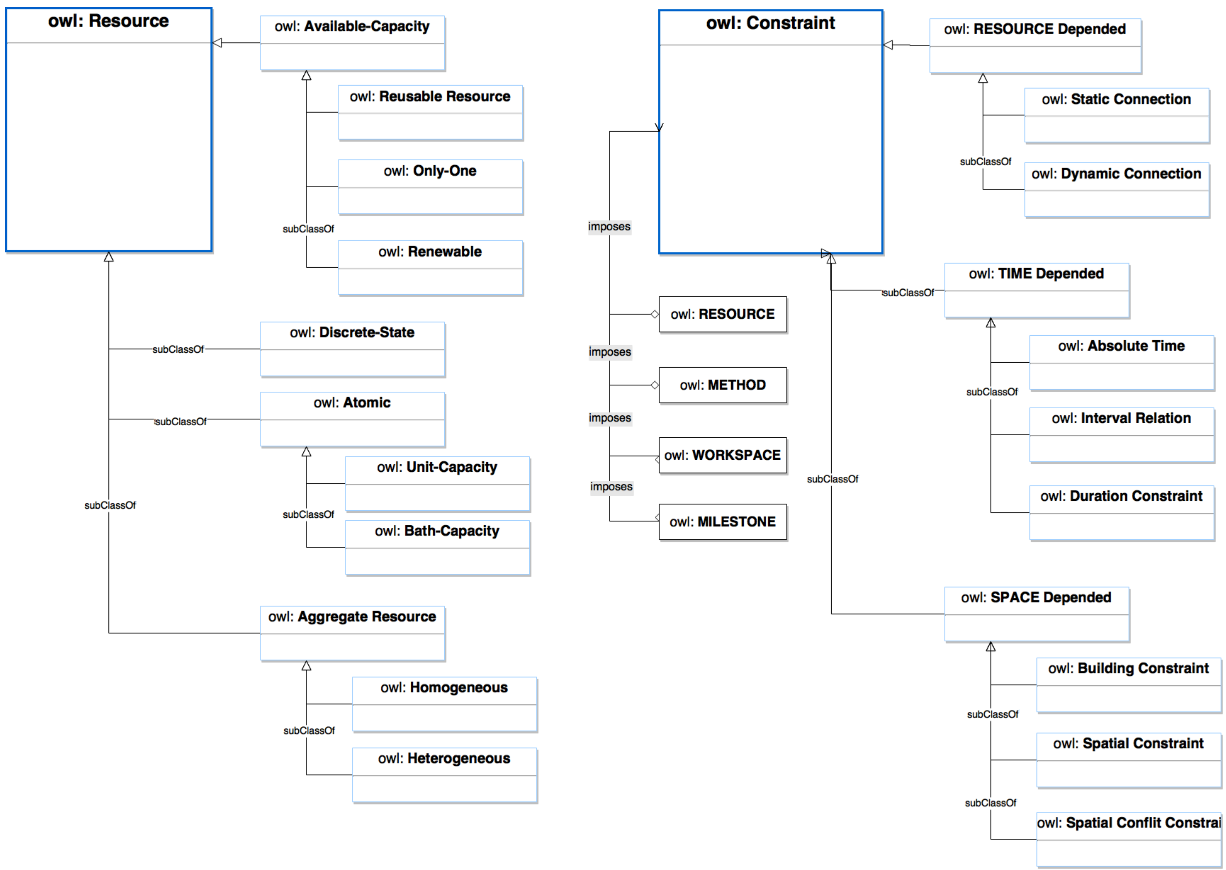


Figure 5-3 Class hierarchy in the construction scheduling ontology: resources types on the left side and constraints types on the right side

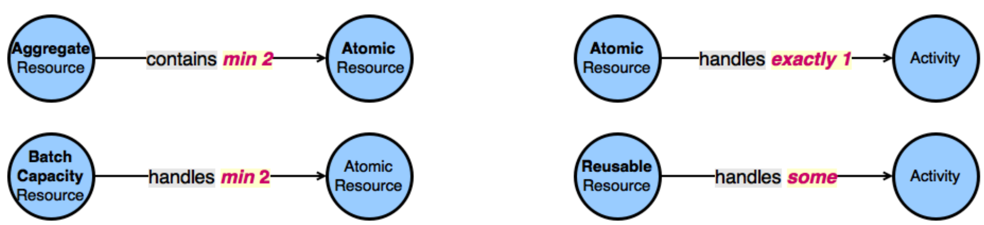


Figure 5-4 OWL restrictions to specify the scheduling ontology as regards to the *resources types*

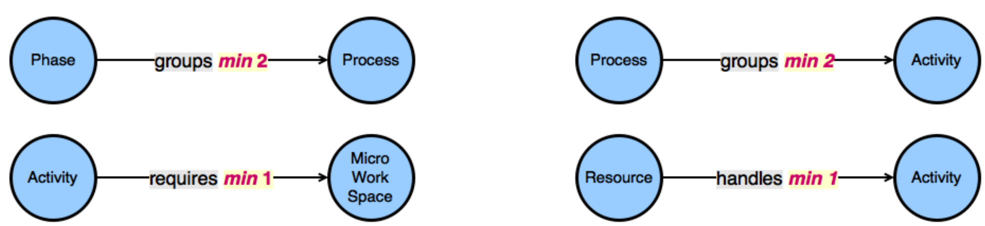


Figure 5-5 OWL classes restrictions as regard the scheduling ontology

Such an ontological model has been converted in a script in OWL language and it is visualized in Figure 5-6 where the graph is automatically generated.

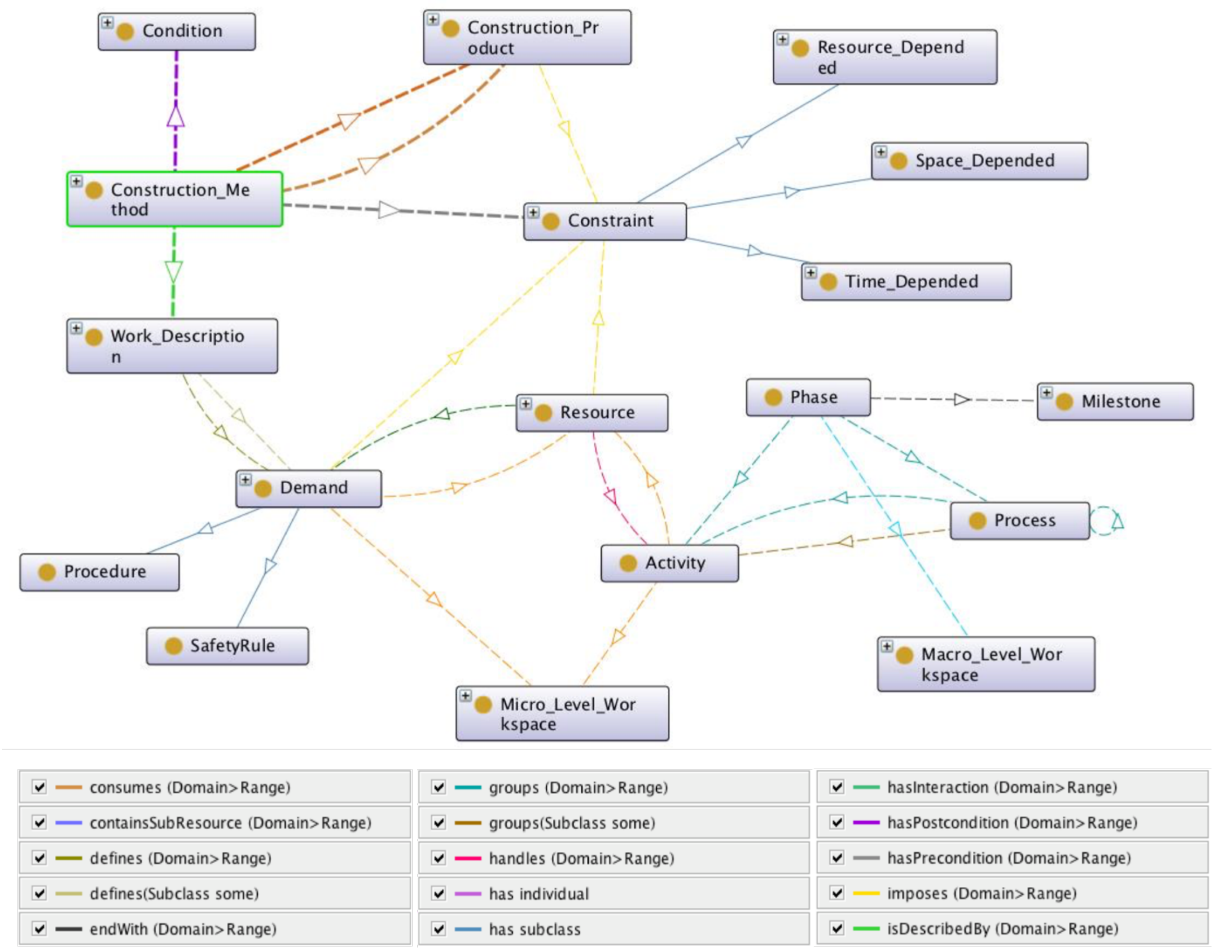
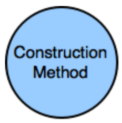


Figure 5-6 Visualization of the dynamic graph representing classes and relations of the Construction Scheduling Ontology. It derived automatically from the script in OWL and visualized in *Protégé*.

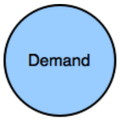
5.4. Specification of the entities

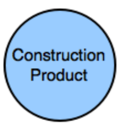
In the table below specifications of classes, relationships and their properties are given.

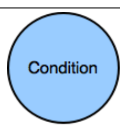
1. OWL Class	**CONSTRUCTION METHOD**		
	**Entity definition**: Abstract entity which describes the Construction Work by means of consumed and produced Work Products and specifies the input goals that drive the ontology. The **Construction Products** are represented by one or more Building Elements [ISO 16739]		
Datatype Properties:	`hasName`**:**	type: ***name*** assignment	
Object Properties:		**Range and Properties**	
	produces:	Domain: ConstructionMethod Range: ConstructionProduct	**FU**
	consumes:	Domain: ConstructionMethod Range: ConstructionProduct	**FU**
	hasPrecondition:	Domain: ConstructionMethod Range: Condition	
	hasPostcondition:	Domain: ConstructionMethod Range: Condition	**TR-AS**
	isDefinedBy:	Domain: Construction Method Range: WorkDescription	

2. OWL Class	**WORK DESCRIPTION**		
	**Entity Definition:** Abstract entity which describes the construction execution, allocation of spaces and required resources for each activity by using generic terms in relation to the construction methods that they are going to use. According to the research object to blend the experienced construction users with the Ontology, the **Method Description** is a generic description.		
Datatype Properties:	`hasDescription`:	type: ***string*** assignment	
Object Properties:		**Properties**	
	isDesribedBy:	Domain: Construction Method Range: WorkDescription	**FU**
	`defines:`	Domain: WorkDescription Range: Demand	**TR**

3. OWL Class	**DEMAND**

	**Entity Definition:** Abstract entities which describes the specific and particular way to perform the description of a **Construction Method** in terms of **Procedures** and **Safety Rules** which are modelled ad sub-classes. The group of outstanding methods at any point determine the scheduling problem to be solved.		
Datatype Properties:	`hasPriorityLevel:`	type: ***real number*** assignment	
	The relative importance of the Demand, bestowing a basis for establishing a partial ordering over the entire set of demands and their procedure.		
Object Property:		**Properties**	
	`imposes:`	Domain: Demand Range: Constraint	**FU**
	`requires:`	Domain: Demand Range: Resource	**FU**

4. OWL Class	**`CONSTRUCTION PRODUCT`**		
	**Entity Definition:** it is a building object. This class comprises all elements that are primarily part of the construction of a building. Construction products are all physically existent and tangible things [ISO 6707-1]. A Product is realized through the execution of some set of Activities. Each product has assigned a Construction Method. For its datatype properties see the *Product* Ontology.		
Object Property:		**Range and Properties**	
	`imposes:`	Class: Constraint	**TR**
	`isConsumedBy:`	Class: Construction Method	**TR**
	`isProducedBy:`	Class: Construction Method	**TR**

5. OWL Class	**`CONDITION`**	
	**Entity definition:** Situation that must be achieved at the beginning (pre-condition) or ending (post-condition) of a **Construction Method**	
Datatype Properties:	`hasName`**:**	type: ***name*** assignment
	`hasObjective`**:**	type: ***string*** assignment
Object Properties:		**Properties**
	`precondition:`	Domain: Construction Method Range: Condition
	`postcondition:`	Domain: Construction Method Range: Condition

7. OWL Class	**`RESOURCE`**

Entity Definition: central to the definition of *the scheduling problem* which represents an entity that supports or enables the execution of a **Construction Method** and its related **Activities**. The resources are considered as a finite supply entity and their availability constraints when and how Activities execute.

Datatype Properties:

`hasName:` type: ***name*** assignment

`hasCapacityLevel:` type: ***real number*** assignment

Each Resource is defined as a limited capacity entity which can be modelled as an integer that indicate how many activities it can handle in parallel at any given time. The Capacity is a quantity of some unit measure (number of workers, volume, surface, weight, etc.) which is available to perform activities over time.

`hasUniformCapacity:` type: ***Boolean*** assignment

It considers Capacity as a scalar quantity and imposes that (Constraint) the sum of the Capacity used by all supported Activities $\leq$ *the Capacity of the considered Resource.*

`hasHeterogeneousCapacity:` type: ***Boolean*** assignment

The slot considers Capacity as the sum of more than one Uniform Capacities

`hasMultidimensionalCapacity:` type: ***Boolean*** assignment

It considers Capacity in terms of more than two quantities and imposes that (Constraint) for each different unit measure the sum of the Capacity utilized by all the Activities $\leq$ *the Capacity of the considered Resource.*

`hasSpeed:` type: ***real number*** assignment

It considers how fast the Resource takes to perform the Activities.

Object Property:

		Properties
isRequiredBy:	Domain: Resource Range: Demand Inverse: Requires	**FU/TR**
imposes:	Domain: Resource Range: Constraint	**FU**
handles:	Domain: Resource Range: Activity	**FU**

7.1 Sub-Class **Spatial Resource**

Entity Definition: A Spatial Resource is a sub-class of a Resource.
This type of resource is time dependent in the sense that their availability for executing the assigned **Construction Method** depends on the available time period of a particular resources and this resource occupies a **Micro-level space** which is a space linked to the execution of a given building object.

Datatype Properties

occupies	Domain: Spatial Resource Range: Macro Workspace

7.2 Sub-Class **Non-Spatial Resource**

Entity Definition: A resource which is time independent and time does not play important role while allocating the **Construction Methods** on the non-spatial resources. The required resource does not occupy neither Macro-level or Micro-level workspace.

7.3 Sub-Class **Available Capacity Resource**

Entity Definition: A Resource whose availability depends on the amount of Capacity that is available. The class contains three subclasses followed described.

7.3.1 Sub-Class **Reusable Resource**

Entity Definition: An Available-Capacity Resource whose *capacity* becomes available for reuse after an Activity to which it has been allocated finishes.
Its main property is the Setup-Duration which specifies how long it takes to configure the considered Resource for another Activity.

	Properties
SetupDuration:	type: ***time duration*** assignment
hasSetupDuration:	Domain: Reusable Resource Range: Temporal Entity

It considers that the Resource requires an amount of time to be reconfigured in the time interval.

7.3.2 Sub-Class **Standard Resource**

Entity Definition: It is an Available-Capacity Resource whose *capacity*, once allocated to an activity does not become available again. Hence in this case the Activity consumes the Resource.

7.4 Sub-Class **Atomic Resource**

Entity Definition: Is the smallest resource which is not divisible and can only support one Activity at a time. This class distinguishes two subclasses listed below.

7.4.1 Sub-Class **Unit-Capacity Resource**

Entity Definition: this Resource can only be used by one **Activity** during any given Time-Interval

7.4.2 Sub-Class **Batch-Capacity Resource**

Entity Definition: This Resource can support more than one **Activity** if three different condition are satisfied:

- it has enough Capacity depending to the supported Activities;
- if the Activities require the same Resource configuration and are temporally synchronized.

7.5 Sub-Class **Aggregate Resource**

Entity Definition: An Aggregate Resource can contain a number of smaller Atomic Resource. Its *capacity* depends on the capacity of its sub-resource and it can be independently allocated to multiple activities in different Time Interval

Object Properties:	*containsSubResource:*	Domain: Aggregate Resource Range: Resource

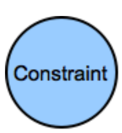
7.5.1 Sub-Class **Homogeneous Resource**

Entity Definition: Is a subclass of an Aggregate Resource composed of more than one sub-resource of the same type.

7.5.2 Sub-Class **Heterogeneous Resource**

Entity Definition: Is a subclass of an Aggregate Resource composed of Resources of different type and capacity.

8. OWL Class **CONSTRAINT**



Entity Definition: a constraint restricts the set of values that can be assigned to a relation between two time intervals in function of different variables according to the scheduling problem. A number of Constraint has been identified which are modelled as sub-classes.

Datatype Properties:	hasApplicability:	type: ***boolean*** assignment
	*Is a value assignment which should be **Hard** or **Soft***	
Object Property:		**Properties**
	isImposedByResouce:	Domain: Constraint

	Range: Resource Inverse of: Resource_Imposes Subclass of: Is_Imposed_By
isImposedByDemand:	Domain: Constraint Range: Demand Inverse of: Demand_Imposes Subclass of: IsImposedBy
isImposedByWorkspace:	Domain: Constraint Range: Workspace Inverse of: Workspace_Imposes Subclass of: Is_Imposed_By
isImposedByConstructionProduct:	Domain: Constraint Range: Workspace Inverse of: Construction_Product_Imposes Subclass of: Is_Imposed_By

8.1 Sub-Class **Resource Depended**

Entity Definition: this class of constraints imposes conditions according to which a *capacity*, assigned to a Resource, is compatible to perform a given Activity.

8.2 Sub-Class **Time Depended**

Entity Definition: A Time-depended constraint restricts the values of temporal decision variables. In this section three types have been included:

8.2.1 Sub-Class **Absolute-Time Constraint**

Entity Definition: An Absolute-Time-Constraint identifies a lower or upper bound on the value of a Time Point which is anchors on a calendar. This constraint borrows Time Points from the following classes: *Method*, *Milestone* and *Schedule Availability Period*.

8.2.2 Sub-Class **Interval Relation Constraint**

Entity Definition: An Interval-Relation-Constraint specifies the relation between two different Time Intervals according with the Time Ontology.

8.2.3 Sub-Class **Duration Constraint**

Entity Definition: A duration-constraint restricts the temporal separation between the Start-Point and End-point of an interval

8.3 Sub-Class **Space-depended Constraint**

Entity Definition: These are constraints which depend on the workspaces allocation within construction site and impose a relation between the time interval which contain those spaces.

8.2.3 Sub-Class **Building Constraint**

Entity Definition: A Building-Constraint imposes a time interval (see *Time Ontology*) between two time intervals referred to the execution of the related construction product. It represents an execution constraint which reflects the structural connection between two Building Objects.
This information comes directly from the BIM data storage and exactly by the classes *IfcStructuralConnection* and precisely *IfcRelConnectsStructuralMember* as specified in Figure 7-4.
For example, if a column has a structural connection with a beam, this means that the beam cannot be built before the column. Hence, the structural connection is rendered by the rule-engine in a Time Depended Constraint which synchronizes the occurrence of the two related Activities Time Intervals (I_1) and (I_2) with an Interval-Relation *Before*.

8.2.3 Sub-Class **Spatial Constraint**

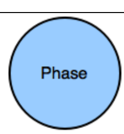
Entity Definition: This constraint maps the spatial relations with occurs between two workspaces for example if a space must be located on a fixed position in relation to another a spatial constraint has been added by the user.

8.2.3 Sub-Class **Spatial-Conflict Constraint**

Entity Definition. A Spatial-Conflict-Constraint define a physical constraint which impacts the assignment of an Interval-Relation-Constraint to Resources and Activities. A physical constraint traduces a conflict detection (as codified in the '*clash report'*) between two workspaces in the ontology.

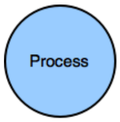
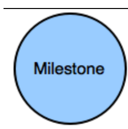
hasName:	type: ***string*** assignment
hasDistance:	type: ***real number*** assignment
hasConstrainedSpace:	Domain: Spatial-Conflict Constraint Range: Workspace

9. OWL Class **PHASE**



Entity definition: a group of strongly-related Construction Methods defined in a given order. A Phase ends with a Milestone.

Datatype Properties:	hasName:	type: ***name*** assignment
Object		**Properties**

Properties:	*groups:*	Domain: Phase Range: Process	**FU**
10. OWL Class	**PROCESS**		
	**Entity definition:** A process represents the most abstract class that groups various activities. If a building components is composed of more than one construction products, the Process is referred to the building components and the Activities to construction products. A sub-process is a Process at a lower granularity level		
Datatype Properties:	hasName:	Type: ***name*** assignment	
Object Properties:		**Properties**	
	isComposedOf:	Domain: Process Range: Activity	
11. OWL Class	**ACTIVITY**		
	**Entity definition:** The Activity is an abstract entity that represents the construction operation related to a Building Objects being installed. An activity requires Micro-Level Workspaces. It is the lower level of a Construction Method that could be formed by an interrelated set of activities. A 1 to 1 ratio should be established between Activity and Construction Product.		
Datatype Properties:	hasName:	type: ***name*** assignment	
Object Properties:		**Properties**	
	requires:	Domain: Activity Range: Workspace	
	hasTemporalEntity:	Domain: Activity Range: Temporal Entity	
12. OWL Class	**MILESTONE**		
	**Entity definition:** A meaningful event. A Milestone involves one or more Work Product. A Milestone represents for instance a Phase finalization		
Datatype Properties:	hasName:	type: ***real number*** assignment	
Object Properties:		**Properties**	
	imposes:	Domain: Milestone Range: Constraint	
	endWith:	Domain: Phase Range: Milestone	

Tab. 2 Specification of entities, relations and properties in the scheduling ontology

Visualization of the ontology computation with all the aforementioned properties is printed in the figure below. It is extracted from the ontology editor Protégé by using the VOWL visualization functionalities.

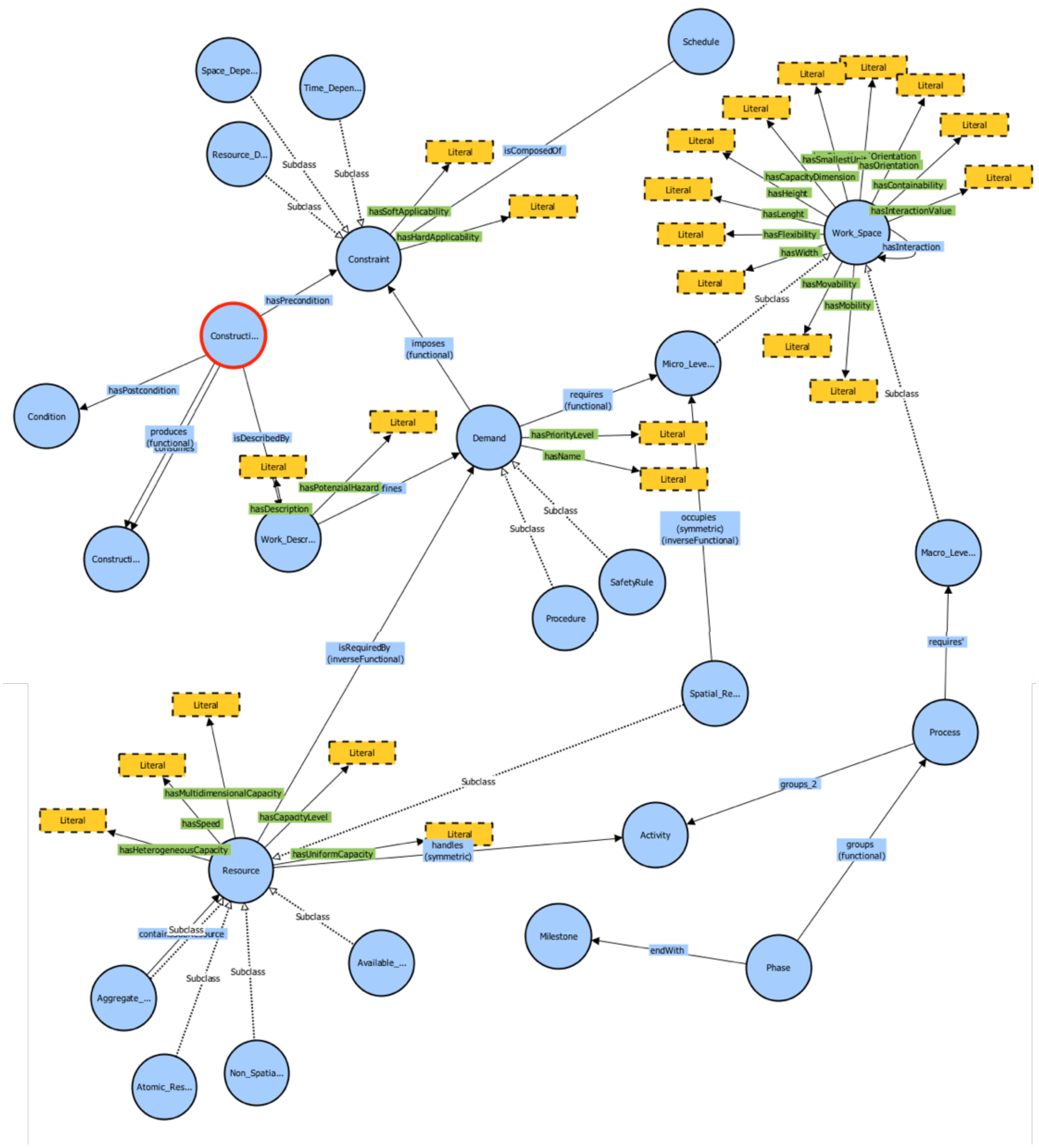


Figure 5-7 Scheduling ontology edited in *Protégé* and visualized with *VOWL* notations (Lohmann, 2014) in a force-directed layout. The graphical representation of OWL entities is made of visual elements. They are: blue circles represent classes and sub-classes, blue rectangles representing property labels of relations have no border to distinguish them from those representing datatypes.

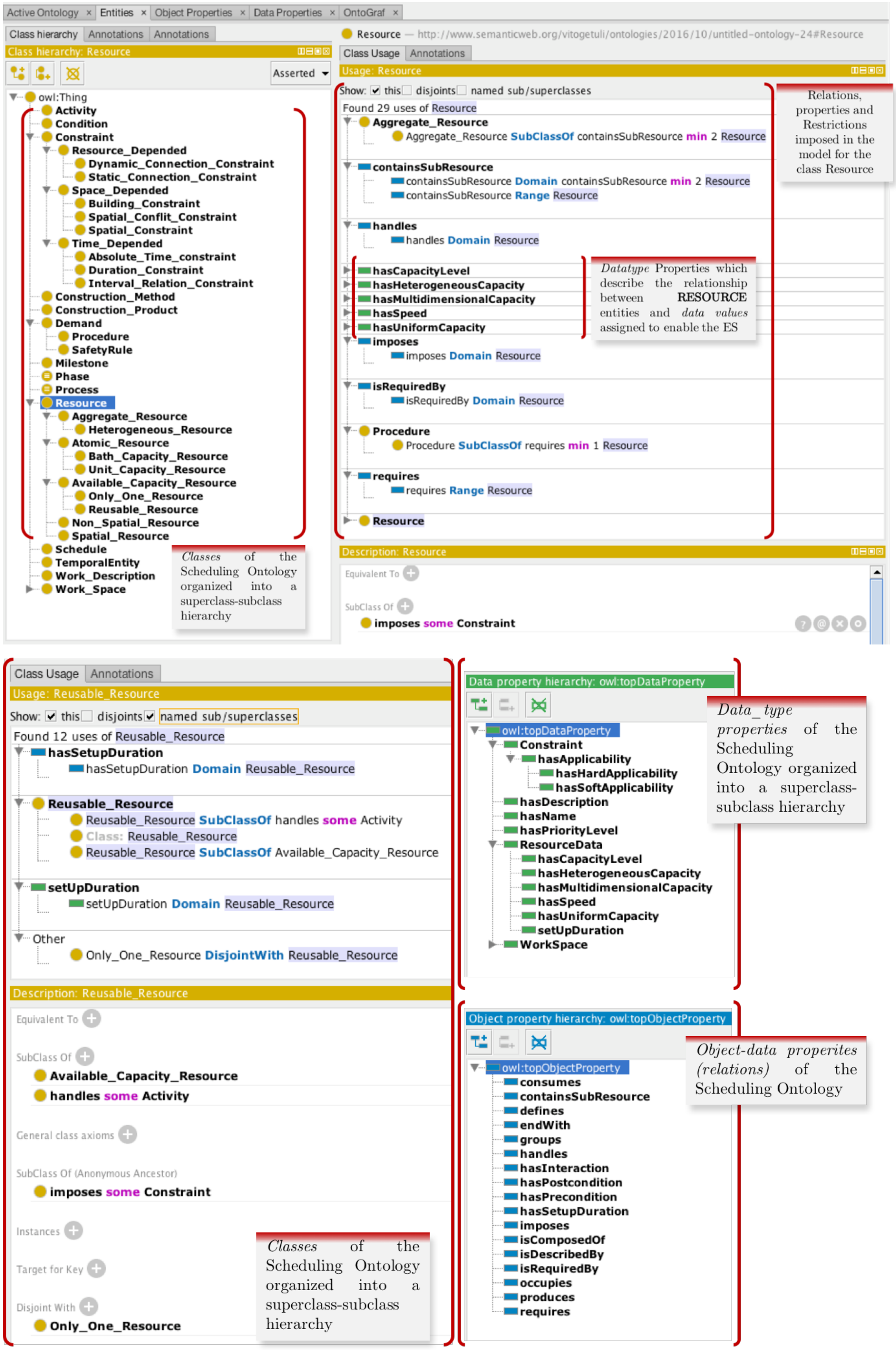


Figure 5-8 Scheduling Ontology implementation and visualization of the proposed data structure. Snippets from the editing environment *Protégé*

Chapter 6 Construction *Space* Ontology

A construction site is a dynamic environment and the workspaces, supporting the execution of each construction activity, are one of the main relevant resource that affects efficiency and productivity of the construction project (Kassem et al., 2015). As a matter of fact, dynamically manage over time workspaces requirements in terms of geometries, locations and interactions with all those spaces that describe the life-cycle workspace evolution associated with the construction activities, is crucial to handle their simulation.

For these reasons, incorporate workspace planning from the spatial and temporal perspectives in the proposed system architecture is crucial, once again by using an ontology-based structure -Space-Ontology-. By doing so, such an ontology is referred to as the knowledge repository which drives the process of generating and allocating workspaces as well as detecting of spatial-temporal conflicts.

The next paragraph describes the in-depth study aimed to structure the *Construction Space Ontology*.

6.1. Specification of modelling objectives

The most challenging work of the Space-Ontology has been to define appropriate default attributes to define workspaces in order to support and manage the following aspects:

(i) **Workspace physical entity**. It aims to define a set of default *Spatial Data* (e.g., dimensions, orientations and positions) to be associated to workspaces that the user should provide the Expert System in order to enable it to activate the built-in algorithm for generating the optimum workspace allocation for each construction method.

(ii) **Workspace structure**. It indicates how workspaces are organized within the site environment in order to define a *Spatial Data Structure* which is the base for describing the spatial relationship between entities based on their geometry locations.

(iii) **Workspace modelling**. This aspect considers the process to automatically submit information (*Spatial Data Modelling*) on point (i) within a BIM Modelling environment, getting over many existing methods which require a manual modelling process that means an extensive user input and working hours that ends up preventing their practical implementation.

(iv) **Workspace analysis and management**. Such an aspect aims to be able to develop an integrated framework with the Knowledge-base which, making use of the above information, should be able to transfer all those that the system requires in order to detect spatial conflicts.

The aforementioned aspects are graphically specified in the figure below (Figure 6-1).

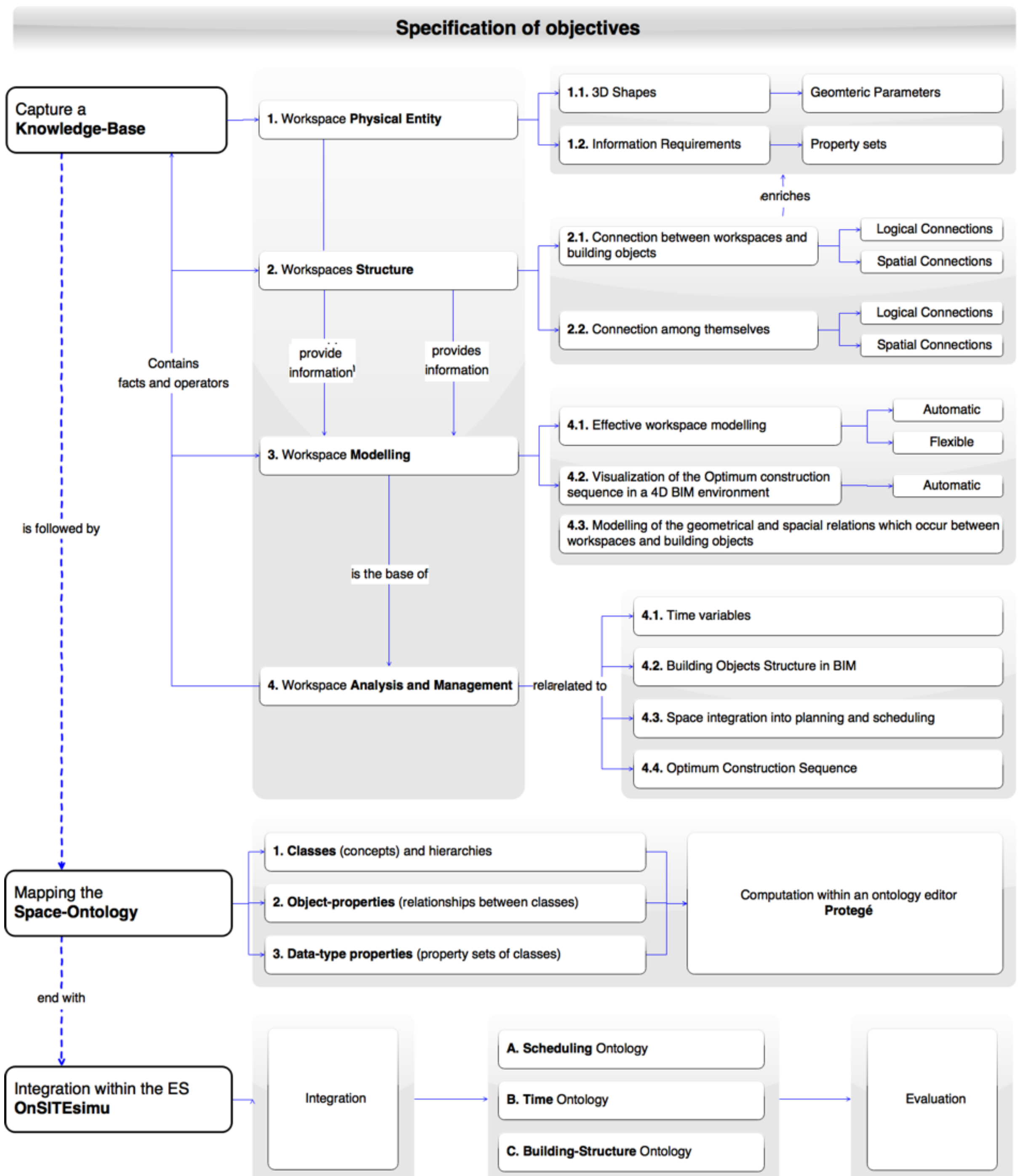


Figure 6-1 Development plan and objectives specification of the workspace ontology

Capture and compute, by using an ontological structure, the concepts in the domain from (i) to (iv) and also the relationships that hold between those concepts is the body of the next paragraphs.

6.2. Overall framework of the Space Ontology

With regard to classifications and descriptions of construction spaces, many research efforts have been made. Each of them propose a different categorization which reflects the relationship between research objectives and workspace management.

On the grounds that the field lacks an authoritative categorization about the construction workspaces, this research extends the categorization proposed by Akinci and Fischer (2000) which, in turn, is an extension of '**C**omponents, **A**ctions and **R**esources (**CAR**)' proposed by Darwiche (1988) and Jagbeck (1994).

Hence, in order to capture the space evolution patterns in the proposed space-ontology the construction workspaces that can bring about physical changes on the construction site are represented as the combination of three categories:

(1) **`Macro-level` spaces**: this category represents the large-scale spaces located across sites in terms of layout areas that are not occupied or required by an individual activity but by a number of activities which is defined as 'Phase' within the 'scheduling ontology' presented in the (*Chapter 5*).

(2) **`Micro-level` spaces**: they represent the spaces required by each stated *construction method* that handles the installation of an 'objects type'. This means that they are spaces located within proximity the building objects to which they refer.

But unlike Akinci and Fischer (2000) spaces occupied by the building components to be installed are not included in this category. This is due to the fact that the proposed space-ontology is a part of an Expert System which is integrated with BIMs and the considered spaces are in addition to all those that are already included in the given BIM which have to be processed.

(3) **`Bound` spaces**: this workspace category, not included in Akinci and Fischer (2000), represents the site boundary and objects that reside on site before the commencement of construction and hence have a known location on site.

That classification is followed up with a more detailed one as shown in Figure 6-2. In this content, it is not important to dwell a great deal on the workspace definitions but on the way, the workspace representation has been dealt within the Expert System.

In this regard, every workspace is not considered to be a static object but a *dynamic entity*, located in a construction site setting, and hence the space ontology, that drives the ES in suggesting the shortest construction sequence, takes account of the concepts listed below:

(1) **Physical Information**: this property set drives the process of workspace generation of 3D shapes and all those other properties such as weight in case of a storage area.

(2) **Topological interaction with other workspaces** included in the same construction method; for example, if a labor crew space requires a protected space on its right, the ontology should capture and manage this relation;

(3) **Topological interaction with building components** to be installed; for example, if a workspace has to be located in a fixed position in relation to the building components to be installed or it doesn't have a fixed preference;

(4) **Reversible properties**: these properties are linked with the shape theory of Cordier and Portie (1989) and include some general properties a workspace should have and therefore the ontology should capture in order to manage the space allocation in site. An example could be the *containability* property which in the case of a crane's volume means that its workspace may contain other spaces and therefore if the 'conflicts checking process' highlights a conflict, the rule-engine should not consider it as constraint for the schedule generation.

This space classification has been computerized within the ontology editing environment in OWL language. Therefore, OWL-classes are interpreted as *sets* that contain 'individuals' specified by the user. They are described using formal descriptions that state precisely the requirements for membership of the class.

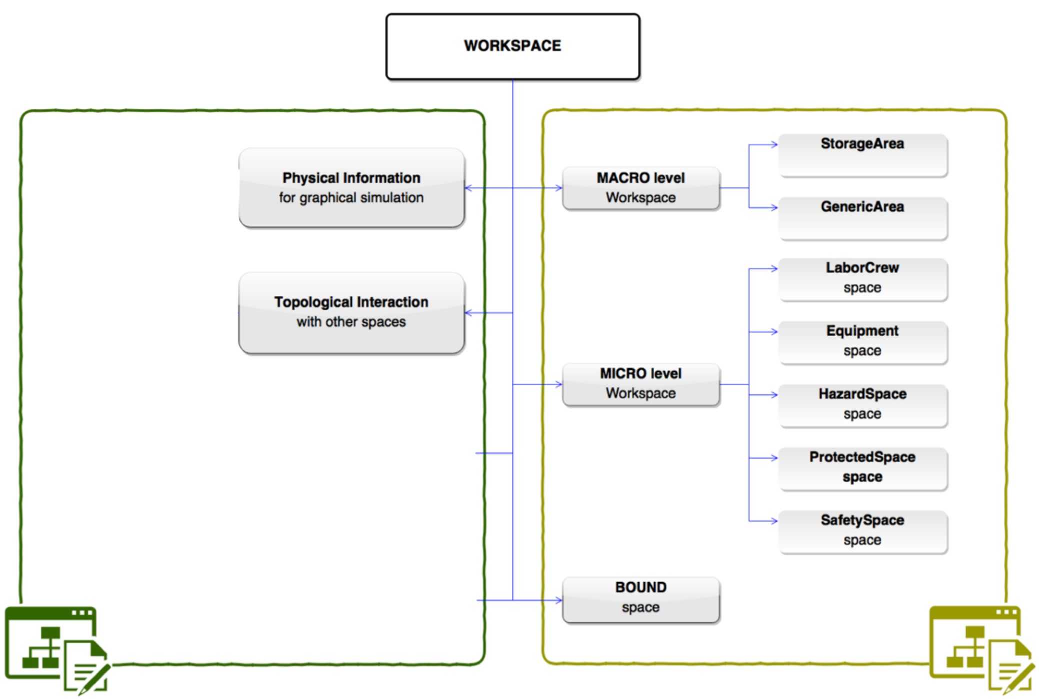


Figure 6-2 Workspaces classifications introduced in the space ontology

For example, the class StorageArea would contain all the individuals that are storage areas in our domain of interest which might be ElectricalMaterial_StorageArea or InstrumentationMaterial_StorageArea, for instance. Classes are organized into a superclass-subclass hierarchy. Subclasses specialize ('are subsumed by') their superclasses. For example, consider the classes MicroLevelSpace and LaborCrewSpace. Labor Crew Space is defined as subclass of MicroLevelSpace. This says that, 'All labor crew spaces are micro-level spaces' and 'All individuals (user-setted) of the class LaborCrewSpace, for every given construction methods, will be automatically members of the class MicroLevelSpace'.

The topological relations of the space ontology and its integration with the other sub domains (scheduling, time and building) are outlined in the next paragraph.

6.3. Topological structure

Here, it is depicted the framework models of the 'space ontology' in terms of classes, properties and relations. For rendering the explication more readable, in body of the text that follows, a different font has been used and the font color is used to distinguish the belonging to one ontology that composes the Knowledge Base.

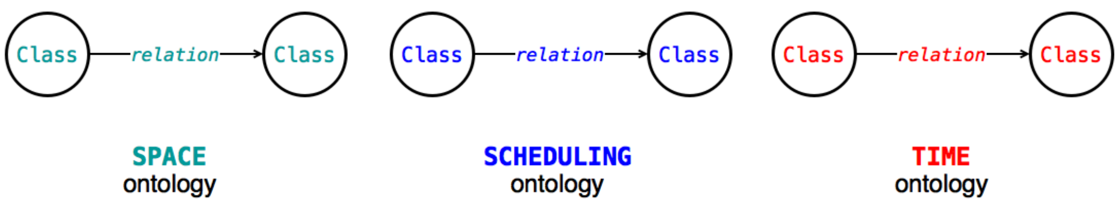


Figure 6-3 Textual notations used in the body of the text to distinguish ontology objects

As before mentioned, core class of the space ontology is ConstructionMethod, the first user interface regarding the ontology compilation with all those data that the user shall be provided for the workspace planning in the prosed system architecture. It means that the user adds *individuals* to all those classes, described below, the ontology is made of. Each construction method *isDefinedBy* a WorkDescription. A WorkDescription *defines* a Demand in terms of Procedure and SafetyRule. A Demand *requires* some Resources. Different classes of Resources are defined according to the Scheduling Ontology but regarding the matter in question, care should be taken on the class Spatial-Resource. A Spatial Resource shall occupy a Micro-Level-Workspace. To manage this relation a *cardinality restriction*[18], which specifies

[18] In OWL language, we can describe the class of individuals that have *at least*, *at most* or *exactly* a specified number of relationships with other individuals or datatype values. The restrictions that describe these classes are known as *Cardinality Restrictions*. For a given property **P**, a *Minimum Cardinality Restriction* specifies the minimum number of **P** relationships that an individual must participate in. A *Maximum Cardinality Restriction* specifies the maximum number of **P** relationships that an individual can participate in. A *Cardinality Restriction* specifies the *exact* number

the *exact* number of relationships that an individual must participate in, is added (Figure 6-4). Instead a *Minimum Cardinality Restriction* is used to ensure that the user will link at least a resource to each construction method.

By using the aforementioned *restrictions* if the user, for example, adds an individual `ColumnInstallation_LaborCrew` that is a member of the class `LaborCrew-Space` and in parallel he doesn't add a `Spatial-Resource` the Consistency Reasoner will suggest an inconsistency or if the user adds an individual of the class `Procedure` which could be, for example, `ColumnInstallation_Procedure` without linking a Resource the reasoner, in the same way, suggests again the inconsistency due to the *Minimum Cardinality Restriction* (`min 1`) which drives the relationship between classes of Procedures and Resources. In this way, the ontology ensures on one hand the given resource is graphically simulated and on the other hand the resource is computed by using the Knowledge base included within the Scheduling Ontology.

Going on in the description, a Resource might *`handle`* one or more `Activities` which, in turn, *`require`* at least one Micro Level Workspaces. Adding a number of individuals, the user may choose how many `Workspaces` handle an activity, with their attached properties.

In such an ontology framework, restrictions are used to specify a `Procedure` requires *at least one* `Resource`, a `Spatial_Resource` occupies *exactly one* `WorkSpace` and so forth as shown in the figure below.

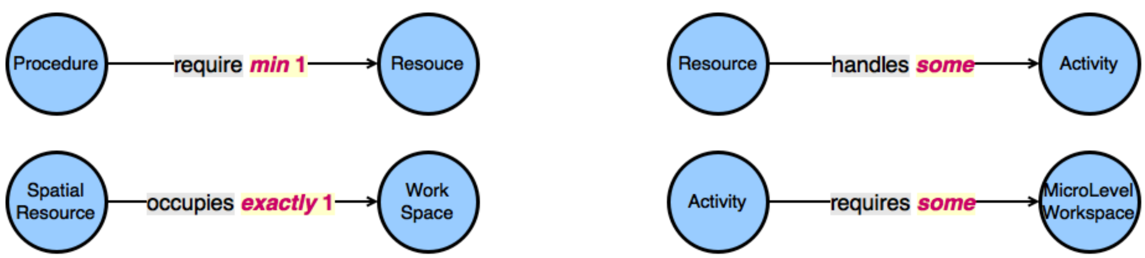


Figure 6-4 OWL restrictions to specify the number of spatial entities assignment required to operate

Different types of workspaces are included in the model by using superclass-subclass relationships. Hence, the class `WorkSpace` contains three subclasses: `Macro-WorkSpaces,` `Micro-WorkSpaces` and `Bounde-Spaces` which represents physical entities in terms of site boundaries and objects that reside on site before the commencement of activities.

Five subclasses support a more explicit description of 'Micro Level' space subtypes: `LaborCrew-Space` represents the space required by a labor crew during the execution of a Construction Method. `Protected-Spaces` are required to protect the construction product for a given time interval. `Equipment-Spaces` identified spaces occupied by the equipment during the execution.

of **P** relationships that an individual must participate in.

In many cases both labor and equipment might `generate` a `Hazard–Space` which is considered different from a `Safety–Space` that represents a safety distance between two workspaces to prevent safety hazards such as collision between two spaces or a tolerance space from objects falling from height or moving in site. The complete class hierarchy consist of classes given in the following figure.

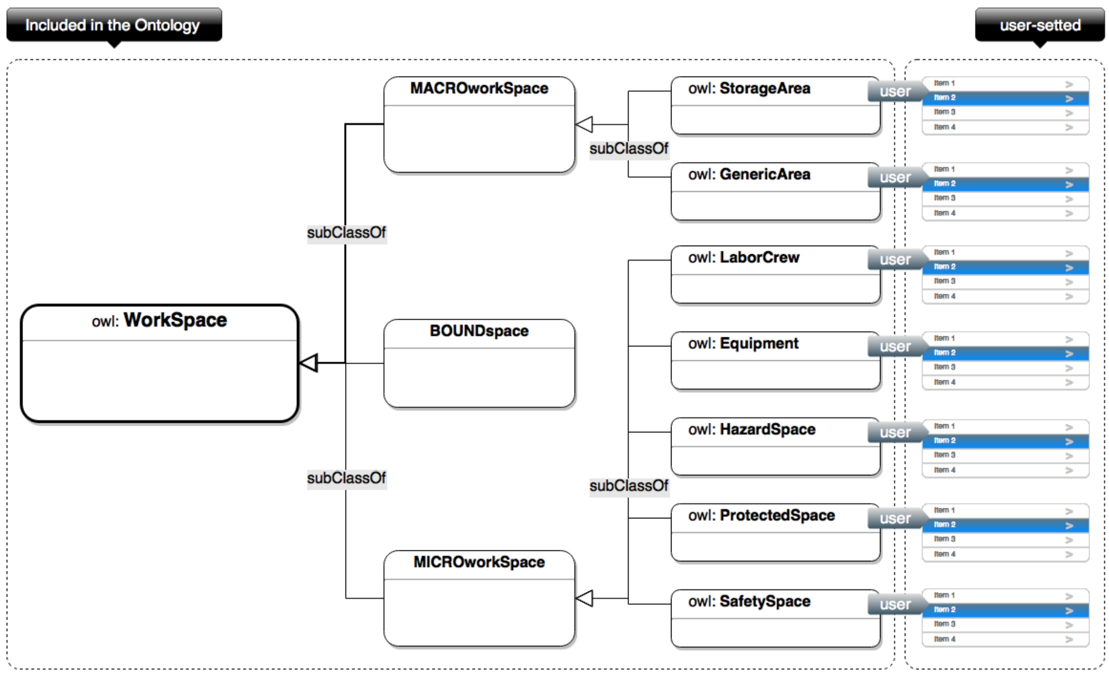


Figure 6-5 Workspaces entities organized in the space ontology in a class hierarchy

The construction of the abovementioned class hierarchy may have seemed rather intuitive so far. However, in OWL *subclass* means *necessary implication*[19].

Each class of `Workspaces` contains four groups of Datatype Properties[20] which support their geometrical and non-geometrical representations. *Datatype properties* link an individual (which in this case means a workspace entity user-entered, e.g. `scaffolding_space`) to a *Datatype* value, in other words, they describe relationships between an individual and data values. The proposed classification is set out in Figure 6-6.

Such an ontological model has been converted in a script in OWL language and it is visualized in Figure 6-7 where the graph is automatically generated.

[19] The fundamental taxonomic constructor for classes is *rdfs:subClassOf*. It relates a more specific class to a more general class. If **X** is a subclass of **Y**, then every individual of X is also an individual of Y. The *rdfs:subClassOf* relation is transitive which means, if **X** is a subclass of **Y** and **Y** a subclass of **Z** then **X** is a subclass of **Z**.

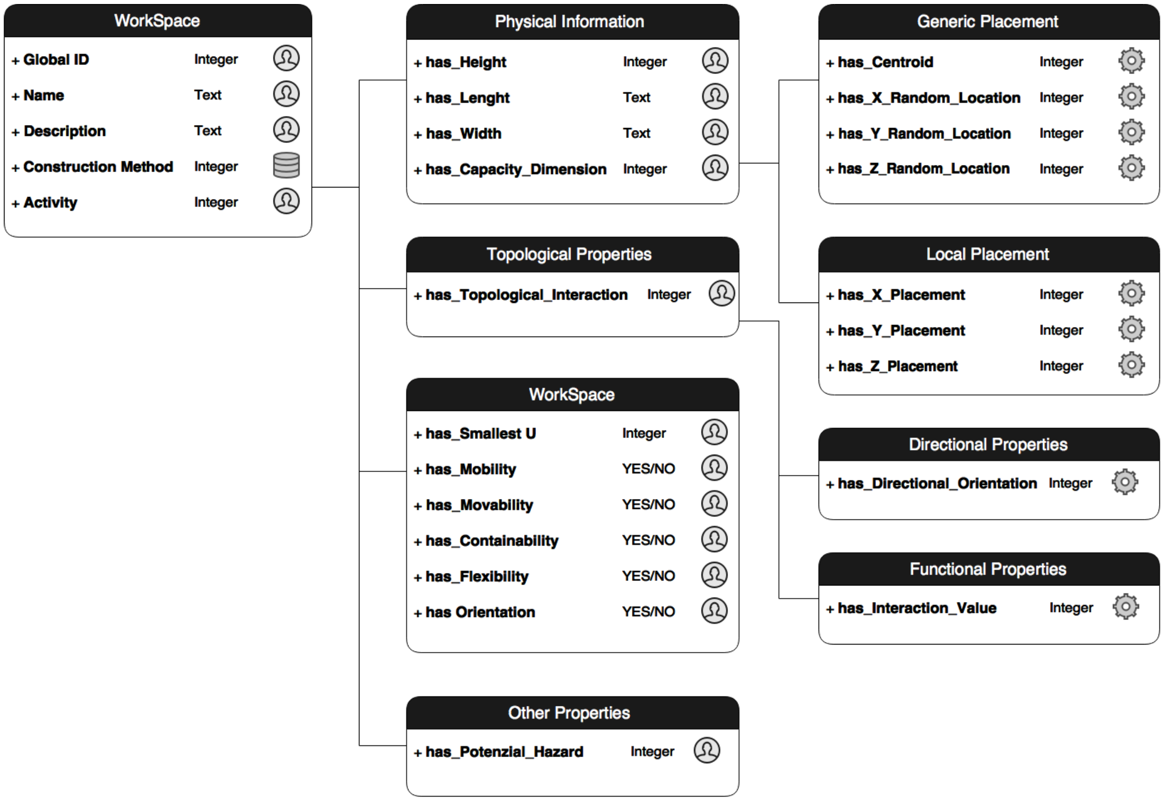


Figure 6-6 Diagram of the property set for representing a workspace within the Space Ontology. This data structure is essential to operate on the KB by using the rule-engine and to communicate such information to the built-in algorithm for the workspaces management.

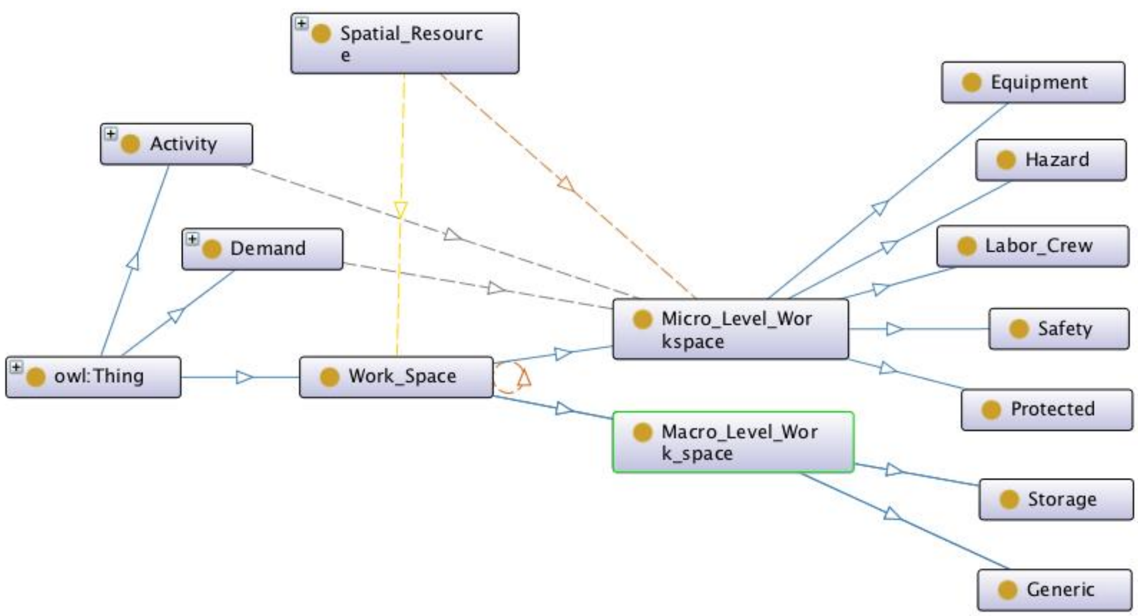


Figure 6-7 Graph visualization representing classes and relations in the Space Ontology, automatically generated from the script in OWL language and visualized by using a *Protégé* functionality.

6.4. Specification of the entities

In the following tabs specifications about classes, relationships and properties are provided.

1. OWL Class	WORKSPACE	
Entity definition:	A workspace represents a physical entity within the construction site. It can be used for various uses according to the proposed sub-classes. The list of properties and motivations for their computation within the system are listed in the tabs.	
Datatype Properties:	`hasID:`	type: ***name*** assignment
	`hasCapacityDimension:`	type: ***real number*** assignment
	`hasDimension:`	type: ***real number*** assignment
	Indicates the dimension of the footprint area which is approximated as rectangular prism which seems to be an acceptable approximation	
	`hasSmallestUnit`	type: ***real number*** assignment
	If the workspace is sizable, this property indicates the dimension of its smallest units. For example, some construction objects such as material are in packages, boxes or other units. When being located on site, they can be located in separated units if occur. The dimension of the smallest unit is needed to decide the size of split location. Object such as equipment are rigid in size and this aspect is reflected in the flexibility property	
	`HasHeight:`	type: ***real number*** assignment
	Includes the highest point of the space	
	`hasCentroid (X-Y axis):`	type: ***real number*** assignment
	Indicates coordinates of geometric centroid of footprint	
	`hasLocation (X-Y axis):`	type: ***real number*** assignment
	Indicates coordinates of workspace location after that the site simulation is carried out and its optimal allocation is defined	
	`hasMobility:`	type: ***boolean*** assignment (YES/NO)
	Indicate if the object is mobile or stationary	
	`hasMovability:`	type: ***boolean*** assignment (YES/NO)
	Indicates if it is acceptable to change the location of object during the project	
	`hasContainability:`	type: ***boolean*** assignment (YES/NO)
	Indicates if the object can be used later to contain another objects inside	
	`hasFexibility:`	type: ***string*** assignment
	Indicated the flexibility of object's shape (flexible/sizable/rigid)	
	`hasOrientation:`	type: ***real number*** assignment
	The angle by which the object is rotated when located on site referred to its reference construction product	
	`hasPotenzialHazard:`	type: ***real number*** assignment
	A value which represents a rough quantification of safety hazards related to the	

activity whose workspaces are simulated

	Properties		
Object Properties:	*imposes:*	Domain: Workspace Range: Constraint	**FU**
	hasTopologicalInteraction:	Domain: Workspace Range: Workspace	**TR**
	Define the interconnectivity level among two workspaces using a set of standardized values.		
	hasInteractionValue:	Domain: Workspace Range: Workspace	
	The Interaction Value defines the proximity level which occurs between two workspace in a range scale from 0 to 10.		
	isRequiredBy:	Domain: MicroSpace Range: Activity	**FU-TR**
	isRequiredBy:	Domain: MacroSpace Range: Phase	**FU-TR**

Follow the list of subclasses that describe all kind of workspaces considered by the proposed ontological model.

1.1 Sub-Class Macro WorkSpace

Entity Definition: spaces located across sites in terms of site-layout requirements. Phases requires a Macro-Level workspaces, Activity requires Micro-Level Workspaces.

8.1.1 Sub-Class StorageArea

Entity Definition: The area required to keep material or tools from the time delivered to site to the time of use.

8.1.2 Sub-Class GenericArea

Entity Definition: Whatever area which the site layout organization requires to ensure Phase progress.

1.2 Sub-Class Micro WorkSpace

Entity Definition: workspaces required by an activity which are located within the proximity of the components (construction products) being installed.

1.2.1 Sub-Class LaborCrew Space

Entity Definition: represents the space required by the labor crew installing the construction product

generates: Domain: LaborCrew Space
Range: Hazard Space

1.2.2 Sub-Class Equipment Space

Entity Definition: represents the space required by the equipment supporting either the Construction Product or the labor crews.

generates: Domain: LaborCrew Space
Range: Hazard Space **TR**

1.2.3 Sub-Class Hazard Space

Entity Definition: represents a hazard space generated by a Labor Crew space or Equipment space

1.2.4 Sub-Class Protected Space

Entity Definition: Represents the space required to protect the construction product for a given time interval.

1.2.5 Sub-Class Safety Space

Entity Definition: Represents a tolerance (safety distance) between two workspaces to prevent safety hazards such as collision between two spaces or a tolerance space from objects falling from height.

1.3 Sub-Class Bound Space

Entity Definition: A bound space is a physical entity which represent the site boundary and objects that reside on site before the commencement of construction and hence have a known location on site. For examples Bound Spaces are trees, existing buildings, marked areas on site such as unenviable, unsafe areas, life line. They occupy space on site and their space is deduced from the total site land. Their Topological Interaction with other workspaces is disjoint and their Interaction Value with other workspaces is 0, the smallest.

Tab. 3 Specification of entities, relations and properties in the space ontology

Visualization of the ontology computation with all the aforementioned properties is printed in the Figure below. It is extracted from the ontology editor Protégé by using the VOWL visualization functionalities.

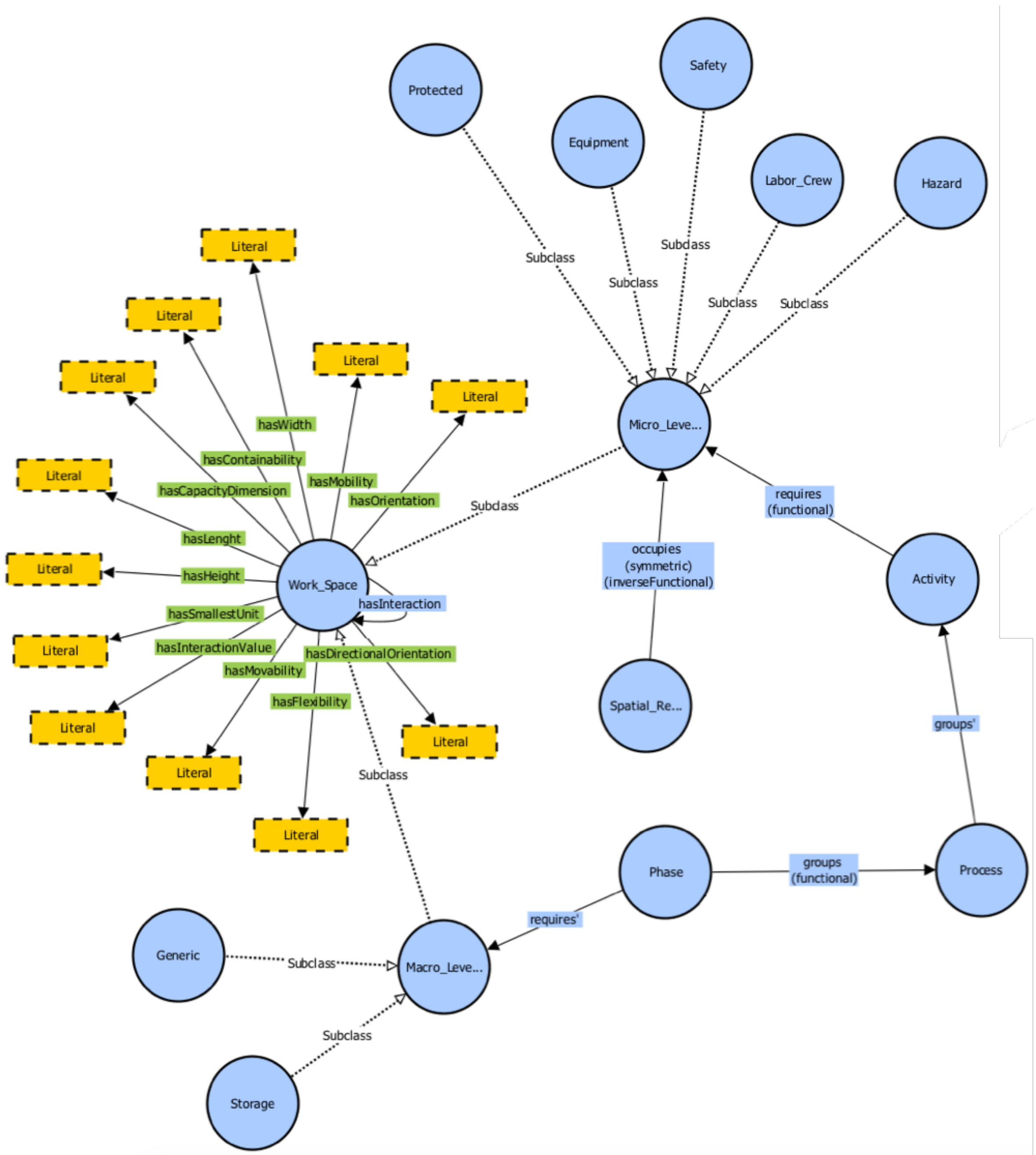


Figure 6-8 Space ontology edited in *Protégé* and visualized with VOWL notations (Lohmann, 2014) in a force-directed layout. The graphical representation of OWL entities is made of visual elements. They are: blue circles represent classes and sub-classes; blue rectangles represent relations and green rectangles represents the property label of data-types.

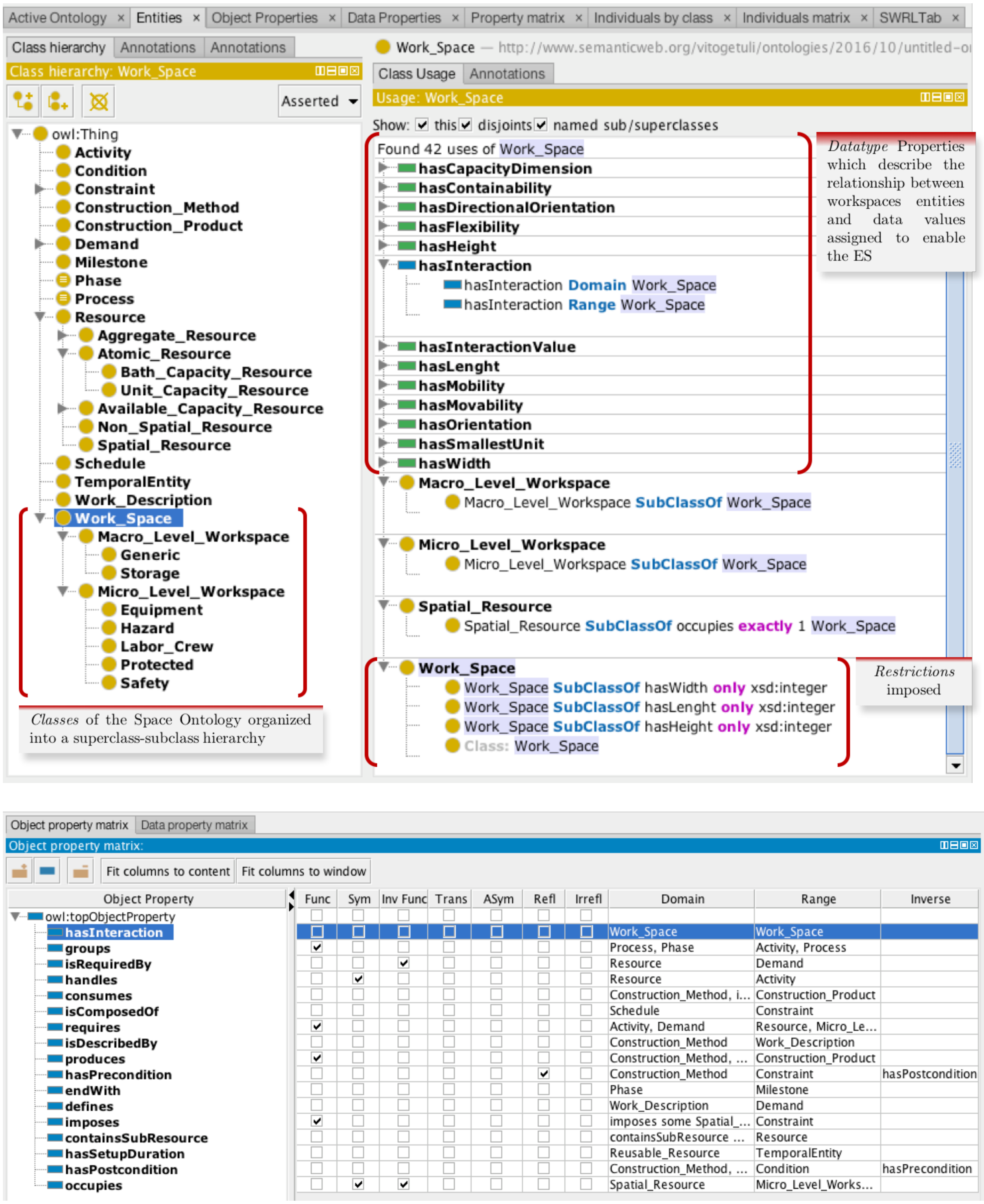


Figure 6-9 Space Ontology implementation and visualization of the proposed data structure. Snippets from the editing environment *Protégé*. On the bottom side, the visualization of relations property matrix with specifications of domains and ranges and properties.

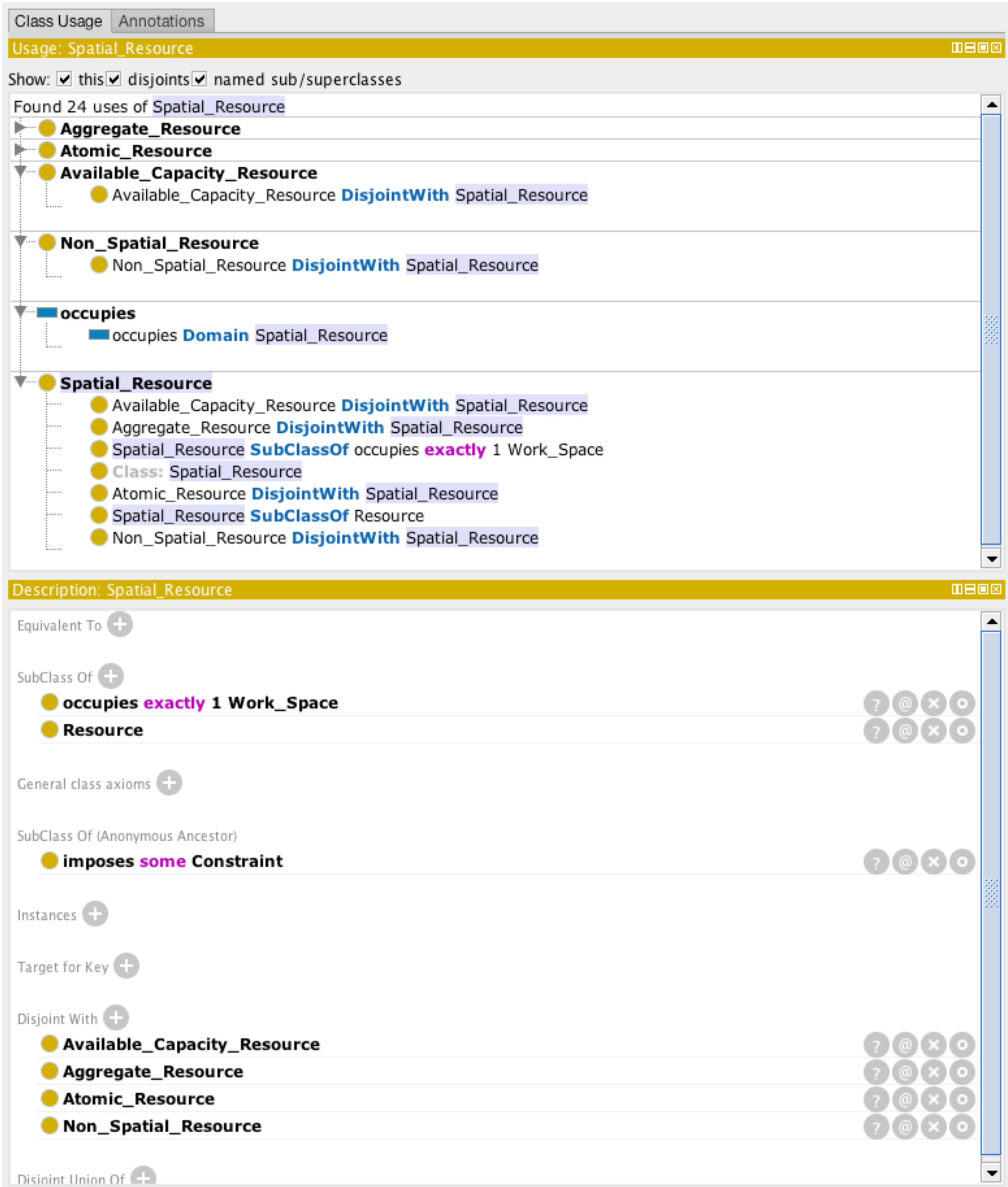


Figure 6-10 Snippet from the ontology editing environment which depicts properties and restrictions of the entity 'spatial-resource'

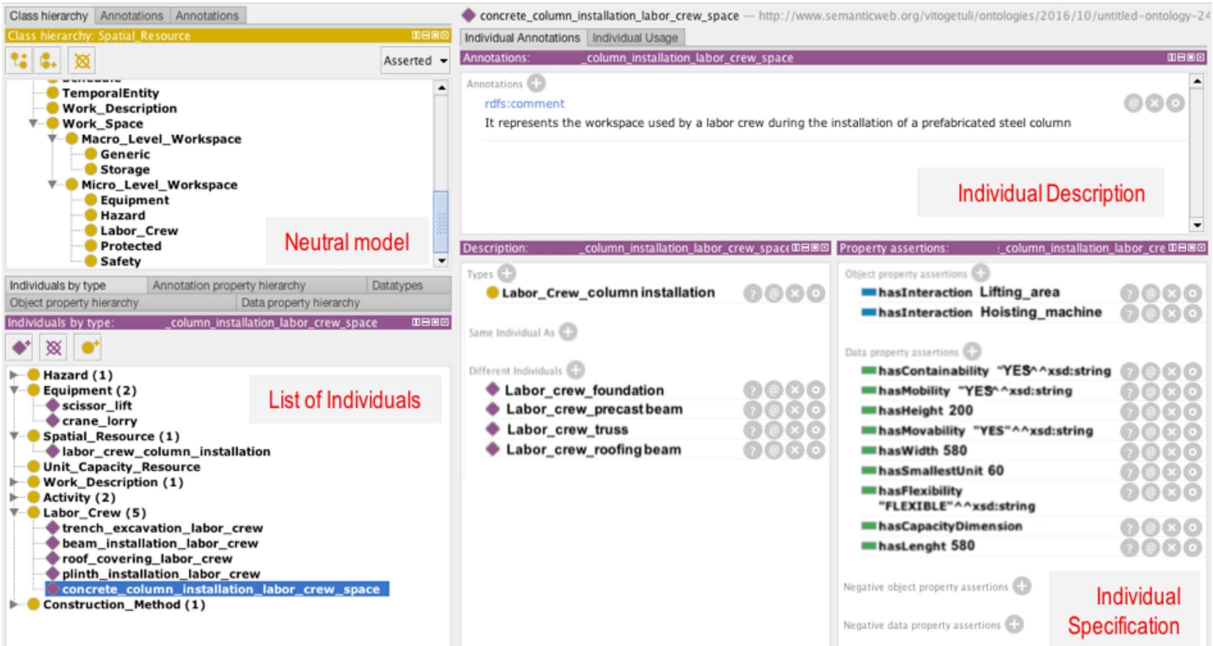


Figure 6-11 Example of individuals included in the space ontology to specify the neutral model

Chapter 7 Construction *Product* Ontology

As already discussed, all Building Information Models (BIMs) are equipped with a standardized interface for data exchange which is nowadays widely accepted in AEC Industry: the so-called IFC (Industry Foundation Classes) standard. The IFC serves as a basis for the exchange of building model data where building components are expressed in terms of objects with attributes. Our approach aims to make use of this information in order to support the knowledge base of the system and the generation of construction schedules.

To achieve this goal different methods have been proposed in literature with the common approach to extract and reuse building data (Dhillon et al., 2014). Unlike in these methods, our approach tries to merge entities and relations, required for the space scheduling purposes, as included in the IFC. This would be desirable in order to define a common data structure ontology-based completely integrated with the other modeling domains before presented (scheduling and space ontologies). This was achieved by using the *ifcOWL ontology* (Buildingsmart, 2014) which is precisely meant to be used to allow extensions towards other structured data sets that are made available using semantic web technologies, (i.e., OWL language) the same one used in our system.

By reason of the fact that the *ifcOWL ontology* is quite complex because of the huge number of classes and properties it contains (Figure 7-1), an in-depth study has been carried out in order to filter only those required entities for achieving our spatial scheduling purposes.

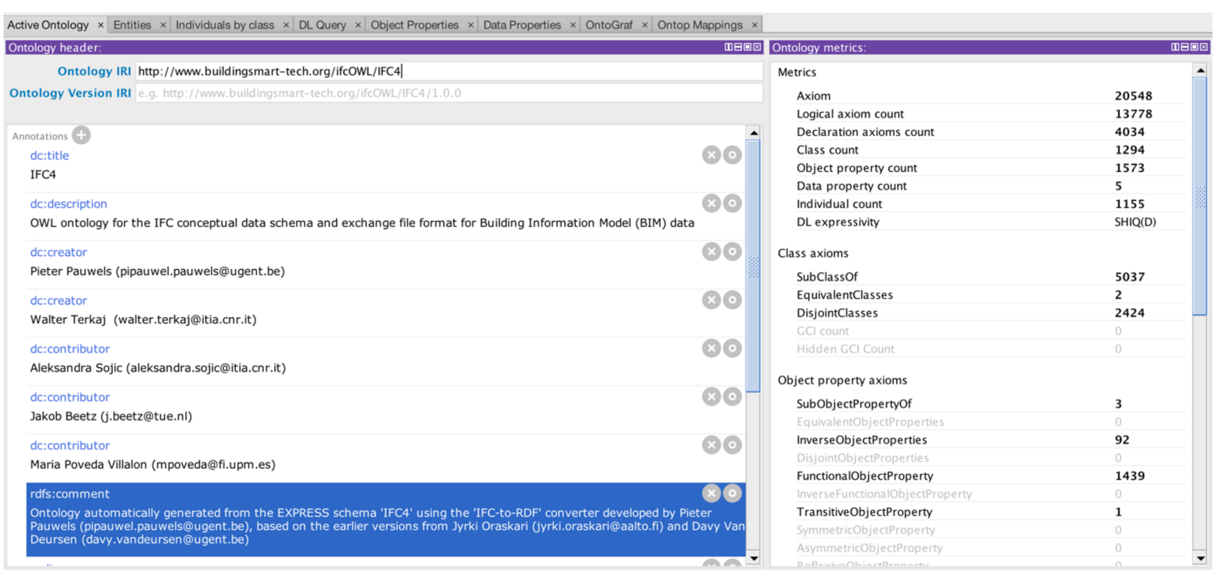


Figure 7-1 Ontology metrics of IFC-schema visualized within the ontology editing environment (*Protégé*)

This approach produced a reduced ontology, here called '*Construction Product Ontology*', for describing the building structure in the proposed system together with required building objects information. In the next paragraph exploration of the IFC structure is provided and then the sub-ontology is presented.

7.1. IFC-based Building Model exploration

Industry Foundation Classes (IFC) represents an object-oriented format which provides a universal base for data exchange in building lifecycle. It has been developed by the International Alliance for Interoperability (IAI) based on the EXPRESS language as a part of the STEP standard [ISO 103030] for the product data exchange. The schema of IFC is quite complex. In order to only represent an IFC building model, for the research scope it is essential to show only the needed objects that formulate a building according to the proposed spatial scheduling algorithm before presented in *Chapter 3*.

The IFC building model is represented with a hierarchical spatial structure which is used to provide a project structure to organize a building projects `IfcProject`. First element within the structure is included in the class of `IfcSite` which defines the area of land on which the project construction is to be completed. A building (`IfcBuilding`) in IFC may have one or multiple stories (`IfcBuildingStorey`). Each building storey may have zero or multiple storeys. Each building storey may have assigned zero or more spaces with certain functions (`IfcSpace`) related to it -i.e., a building structure which has only one wall is a building with zero spaces-. For example, rooms in IFC are represented by the `IfcSpaces` class with a predefined *`PropertySests`*. Building elements and opening elements are represented as subtypes of spatial structure elements (`IfcSptailaStructureElement`). Each building element (`IfcBuildingElement`) has zero or more opening elements (`IfcOpeningElement`) i.e., a wall without any door or window has zero openings, whereas each opening element (like door, window) is attached to only one building element. `IfcSpatialStructureElement` links between building elements and upper structure of building (project, site, building, storey and space) as it defines spatial structure of a building and its parts. Each `IfcProduct`, that is an abstract representation of any object that relates to a geometric or spatial context, is located in `IfcGrid` which is a planar design grid defined in 3D space used as an aid in locating structural and design elements. The position of the grid (ObjectPlacement) is defined by a 3D coordinate system. The relative placement of a product in relation to the placement of another product or the absolute placement of a product within the geometric representation context of the project is defined by the class `IfcLocalPlacement`. The particular geometric representation of a product is defined by `IfcShapeRepresentation` which includes several *`RepresentationIdentifier`*. It is derived from refers to geographic locations (IfcLocalPlacement) of building elements and their geometries. Geometric representation in IFC is built on solid geometries.

7.2. Topological structure

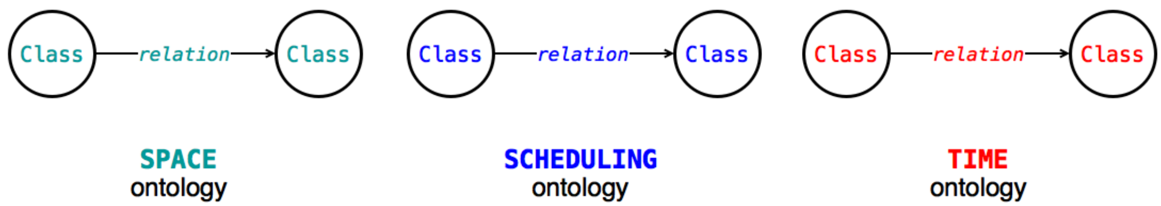


Figure 7-2 Textual notations used in the body of the text to distinguish ontology objects

According to both spatial-scheduling algorithm (*Chapter 3*) and IFC exploration, the selected entities and properties able to support the system in working on the exploration of the shortest construction sequence are listed below and graphically depicted in (Figure 7-4), moreover the main topological relations with the other sub-ontology are specified:

(a) IfcBuildingElement. This abstract class, that works as super-type, comprises all elements that are primarily part of the construction of a building, i.e., its structural and space separating system. Sub-types are IfcBeam, IfcColumn, IfcCurtainWall, and so forth. The selected classes included in the knowledge base of the presented system architecture are graphically represented in (Figure 7-3). They are defined as sub-classes of the master-class ConstructionProduct. In this way, for each of them a different ConstructionMethod will be define which in turn will consist of a number of individuals as grouped within classes (i.e., LaborCrew–Space, Protected–Space, Equipment–Space, etc.) and properties described in the Construction Space Ontology. By doing so, each building objects, the given BIM will be composed of, will be allocated to a specific construction method via isProducedBy transitive relation.

(b) Then, within the IFC-based data structure, each element is specified by using a number of capabilities, mainly through *property sets* (e.g., Material, classification, documentation, boundary, coverings, etc.). For our scheduling purposed in order to provide the system with information able to support the rule-engine to automate the generation of the structural construction sequence, the objectified relationship IfcRelConnectsElements has been selected. It handles the connectivity between elements. It is a 1 to 1 relationship. The connectivity may be related to the shape representation of the connected entities by providing a connection geometry or a point connection with attributes with assigned values on X, Y and Z-axis.

If such a relation exists between two building objects, making a comparison between the height attribute on Z-axis, the rule-engine can establish a time relation *IntervalBefore* between the items with a higher Z-value and the other one.

(c) Furthermore, according to the spatial planning purpose a reduced BIM, which includes space availability in site -building objects and workspaces- should be generated in order to:

- simplify the geometries representation of building elements and
- find the optimal workspaces allocation in reference to the spatial allocation of such building elements to which these spaces relate.

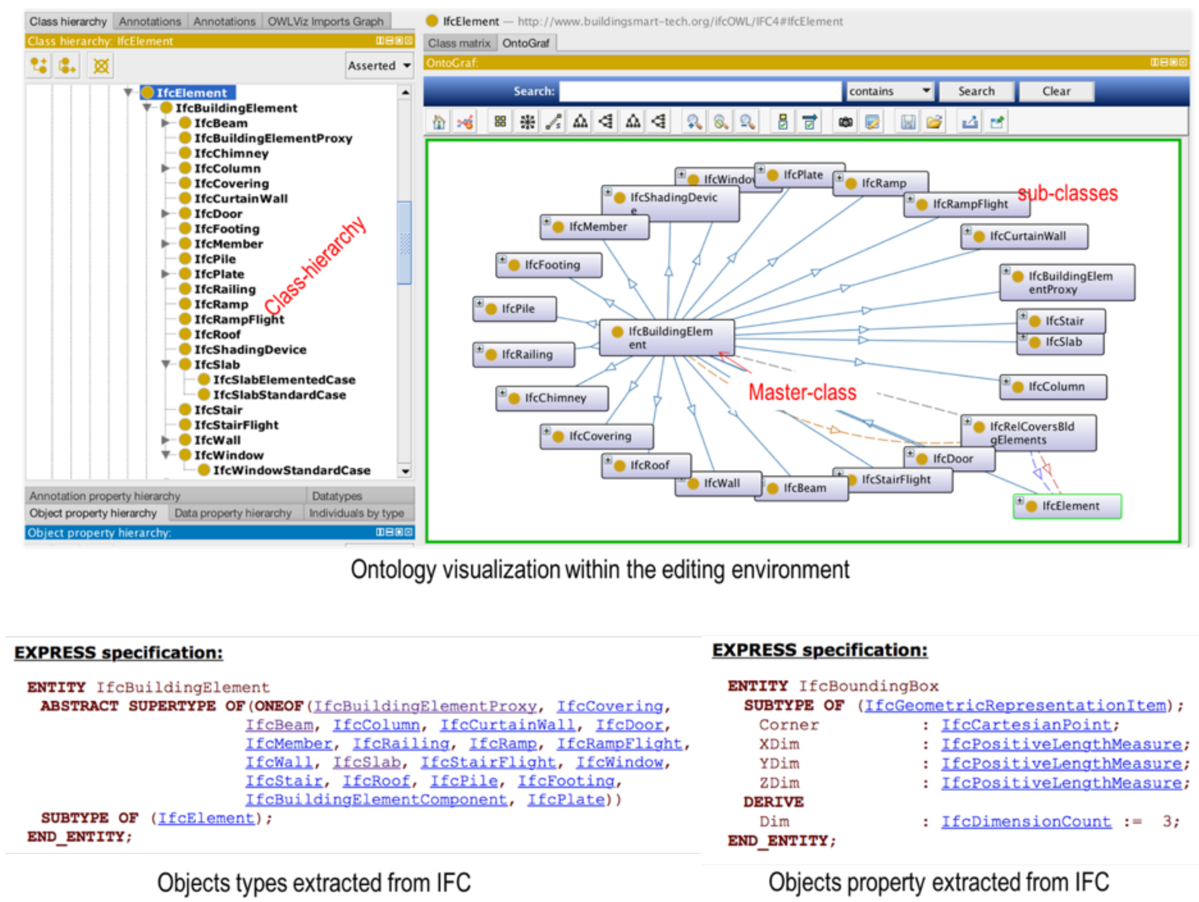


Figure 7-3 Typologies of building objects considered in the System as imported from the IFC structure. Snippet from the ontology editing environment Protégé

Therefore, any `IfcBuildingElement` will be represented as a *bounding box*, which shows the maximum extend of the body within the coordinated system established by the attributes of the `IfcLocalPlacement` class. The bounding box representation is the simplest geometric representation available. It is defined by a Corner being a three-dimensional Cartesian point and three length measures defining the X, Y and Z values of the box as depicted in point (**2**) and (**5**) of Figure 7-4. By using such information, the *built-in algorithm* integrated in the proposed system architecture automatically sculpts geometries of the building elements in the right spatial allocation in site (specifications in *Chapter 11*).

The aforementioned OWL entities have been included in the script in Protégé to settle the definition of the Knowledge Base on which the rule-engine will operate on.

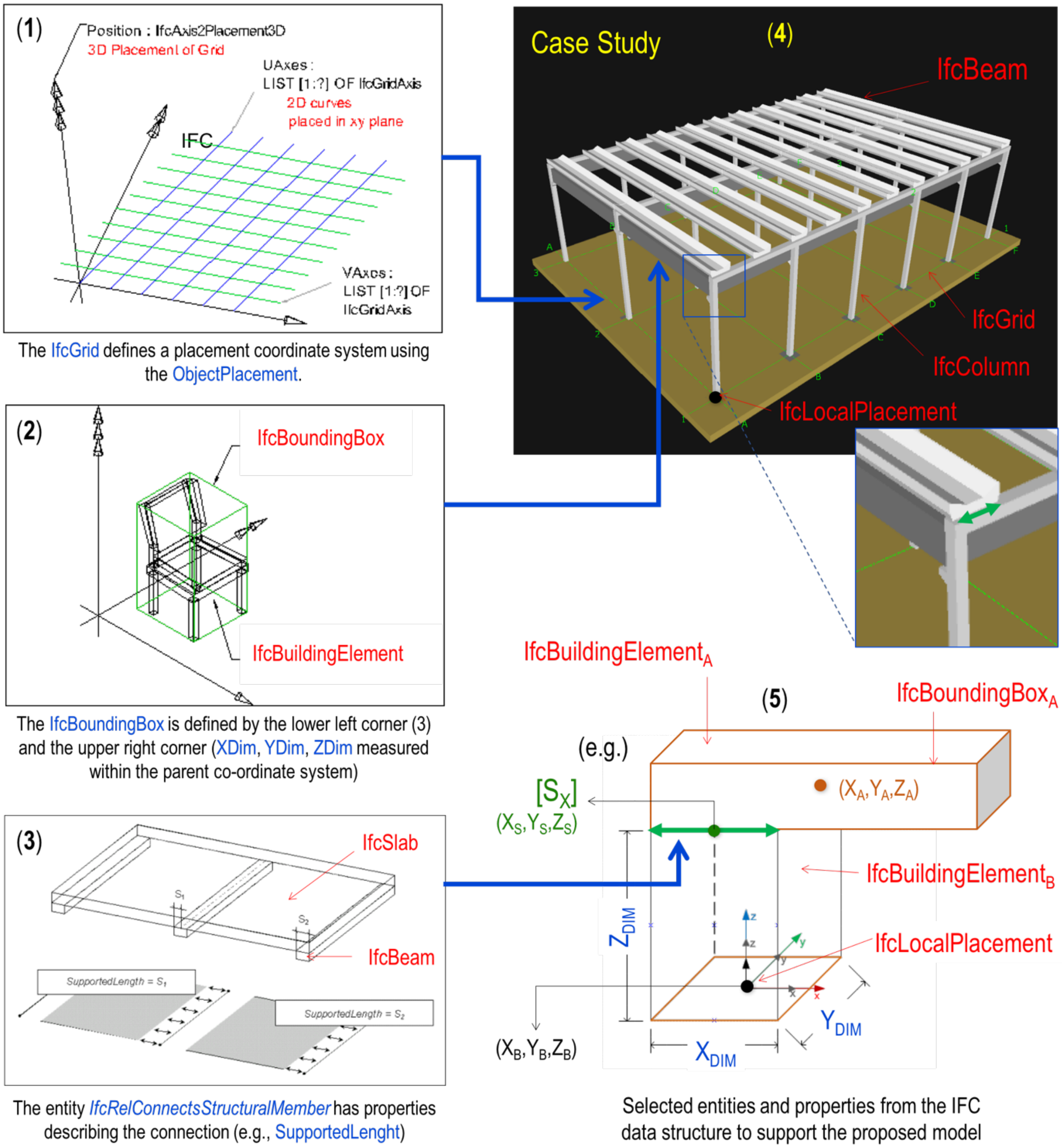


Figure 7-4 Selected entities and properties, graphically presented, able to support the system in working on the exploration of the structural construction sequence by using data coming from the IFC structure

7.3. Introduction of BIM data in the KB

In specifying the Knowledge-Base, two groups of information are considered: building data from a given BIM according to aforementioned specifications, and user's experience acquisition. Their integration in the knowledge base by means of ontologies –OWL individuals-, has been designed as graphically depicted in the figure below (Figure 7-5) and later specified.

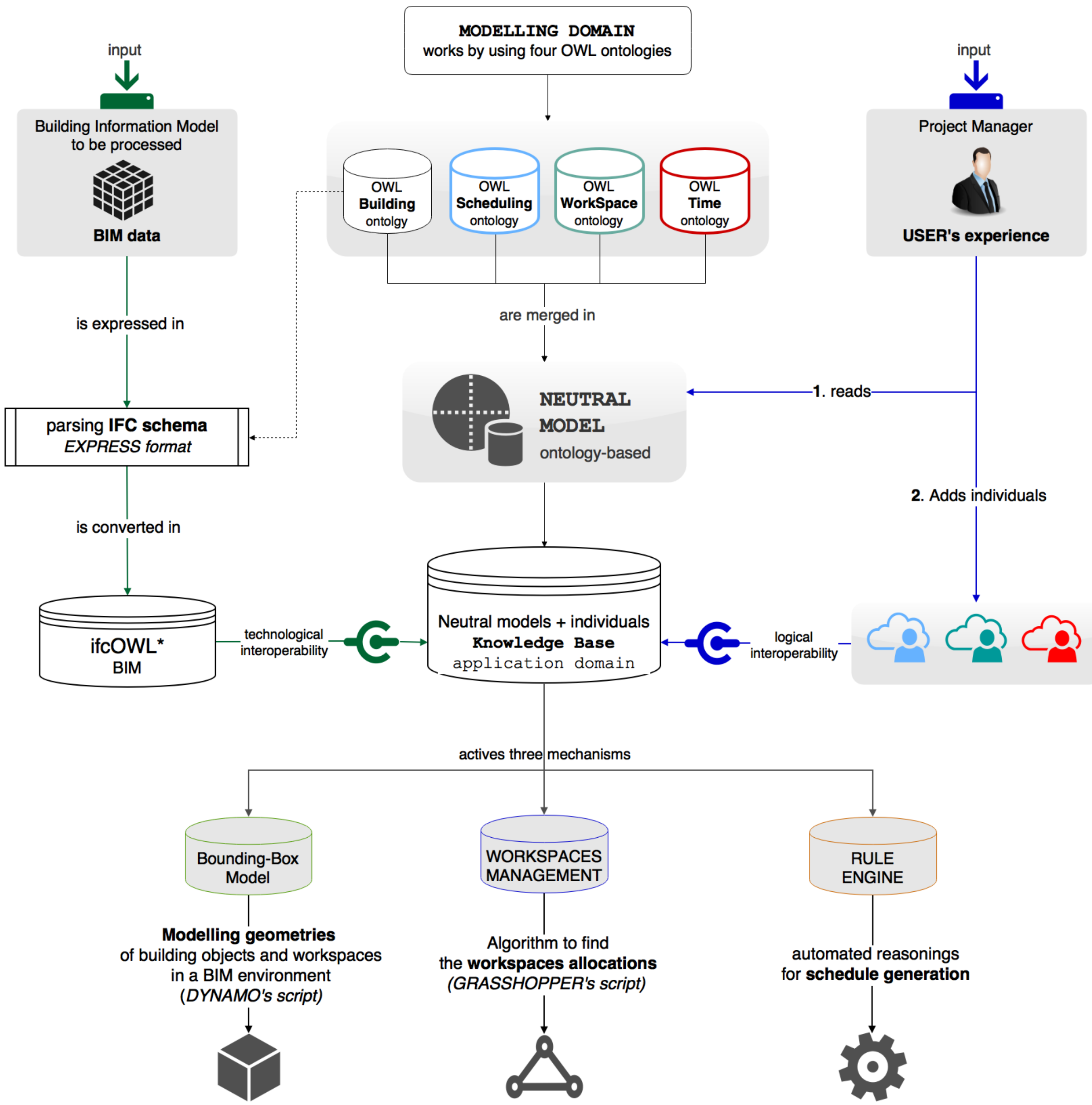


Figure 7-5 Designed information-flow implemented to introduce BIM data (as filtering according to the Product ontology) and user experience within the Knowledge-Base

Step (1): The configuration process starts having a Building Information Model to be processed, whatever the Level of Development (LOD) is made.

Step (2): The given BIM, with parsing IFC schema defined in EXPRESS format (ISO 10303-11:20041)[21], is converted in OWL format (later called *ifcOWL**). So that means that the expert system has at its disposal the BIM in an ontological structure (classes, relations, properties, individuals).

Step (3): The ifcOWL* is merged with our 'construction scheduling ontology' so as to capture only ontological components needed (i.e., geometry information of building objects with their bounding boxes and local placements, structural connections among objects). This process is carried out by using a functionality of the editing environment *Protégé*. In this way OWL-individuals generated using BIM project information.

Step (4): The user, at this point, can add individuals to the ontology structure in terms of construction methods (specifications at *Scheduling Ontology* in *Chapter 5*) and their workspaces with their related property sets (specifications at *WorkSpace Ontology* in *Chapter 6*. It should be noted that, at the beginning of the configuration process, the 'construction process ontology' works as a *neutral model* which contains no specific object but only abstract entities.

Step (5): At this point, the reasoner updates the ontology and the Knowledge base is uniquely populated with specific individuals and it will be ready to support reasoning mechanisms.

---

[21] ISO 10303 specifies a language by which aspects of product data can be defined. The language is called EXPRESS. ISO 10303-11:2004 also specifies a graphical representation for a subset of the constructs in the EXPRESS language. This graphical representation is called EXPRESS-G. EXPRESS is a data specification language as defined in ISO 10303-1. It consists of language elements that allow an unambiguous data definition and specification of constraints on the data defined.

Chapter 8 Construction Time Ontology

The representation of temporal relationships and temporal properties in the proposed model is a crucial point mainly because a schedule describes the construction process across time. In this research, we are concerned with exploring temporal relationships between ontological entities. Therefore, we will adopt a discrete, linearly-ordered time domain, and we will focus on absolute time. The specific time relations we are interested in are those of Allen's interval algebra (Allen and Ferguson, 1997). We incorporate the time dimension into the model by associating time intervals to relationships between entities. We have customized the time ontology presented in (Cox and Little, 2016) after reviewing the most used by other authors. In the following paragraph its structure is presented.

8.1. Topological temporal entities

Four key points have driven the modelling of time variable in the knowledge-base. They aim to manage the time progression of entities in site and the translation of spatial constraints (e.g., workspaces conflict, etc.) into temporal constraints (time interval between conflicted entities) to impose into the schedule generation:

- **(i)** Temporal Relations between entities included in the other sub-ontologies;
- **(ii)** Representation of time positions;
- **(iii)** Duration of construction intervals;
- **(iv)** Temporal Reference System of the Expert System.

By doing so unlike traditional Gantt Chart and network diagrams, temporal relationships with properties can be established between two entities and moreover each entity can be linked to more than one entity at the same time and with different relationships types.

Therefore, as it pertains to (i), the `TIME ONTOLOGY` is based on binary relations on intervals in order to represent temporal information on which a schedule may be structured and to address the problem of automatic reasoning on such information. This is carried out in the ontology by using the class `TemporalEntity` which has object-properties able to assign temporal instants to the individuals (e.g., building objects, workspaces, etc.) in order to define its beginning and end (i.e. *hasBeginning* and *hasEnd*). Therefore, this class defines the time properties of each `Activity` which, in turn, is related with other classes, i.e., `Resources` and

Workspaces via the relationships *handles* and *requires* respectively. Such a network allows the ontology to automatically assign the same temporal interval of an Activity to Resources and Workspaces which handle the Activity itself when a time interval has been assigned to only one of them. This is achieved assigning the transitive property to the object-properties *hasBeginning* and *hasEnd*.

Interval and Instant are two subclasses of TemporalEntity. This specification allows the system to draw a distinction between Milestone and Activity in the Schedule, indeed, a Milestone is an Instant, as an interval with zero length, and an Activity is an Interval. The object-property *isA* is used to define these logical connections.

ProperInterval is a subclass of Interval and is used to define possible binary connections between two intervals (i.e. *intervalMeets*, *intervalOverlaps*, *intervalBefore*, *intervalDuring* etc.), and therefore between to Activities and its related classes, (e.g., resources and workspaces). The specific relations interested are those of (Allen and Ferguson, 1997).

Such classes are the most important for the construction schedule generation due to the fact that they provide the rule-engine (see the following chapter) with new relations to establish between all those entities somehow included in the 'conflicts checking process'.

As it pertains to (ii), three classes describe temporal position within a reference system and all have an object property hasTRS to indicate the temporal reference system TRS (Temporal Reference System). TimePosition has properties to describe the position using both a number (i.e. a temporal coordinate), or a nominal value.

As it pertains to (iii), the duration of an interval can have different descriptions: class Duration describe the duration as a number. GeneralDurationDescription has different properties to specify a duration (e.g. hours, days, months, etc.) and DurationDescription fixes the temporal reference system used in the proposed ES to the Gregorian calendar, meeting requirements. Specifications of classes with their interaction domains and range as computerized by using OWL language in *Protégé* are described below in Tab.4 and graphically depicted in Figure 8-1.

8.2. Specification of entities in the Time Ontology

Below, the structure of the Time Ontology (Cox and Little, 2016) which drives the *OnSITEsimu* in defining (1) *temporal relations*, (2) *temporal reference systems*, (2) *time position* with time unit and (4) *interval duration* is presented.

T1. OWL Class	**TEMPORAL ENTITY**
Entity definition:	This Class define a temporal interval which is assigned to each Activity.

Object Properties:		**Properties**
	before:	Domain: Temporal Entity Range: Temporal Entity
	If a temporal entity T_1 is before another temporal entity T_2, then the end of T_1 is before the beginning of T_2.	
	after:	Domain: Temporal Entity Range: Temporal Entity
	If a temporal entity T_1 is after another temporal entity T_2, then the beginning of T_1 is after the end of T_2.	
	hasBeginning:	Domain: Temporal Entity Range: Instant
	hasEnd:	Domain: Temporal Entity Range: Instant
	hasDuration:	Domain: Temporal Entity Range: Duration
	This object property fixs the duration of a Temporal Entity expressed as a nominal value	

T2. OWL Class	**INTERVAL**	
Entity definition:	This class define a Temporal Entity with an extent or duration	
Object Properties:		**Properties**
	inside:	Domain: Interval Range: Instant

T3. OWL Class	**INTERVAL RELATION**	
Entity definition:	A Temporal Entity with a defined duration by using the class Interval which is supported by the seven interval relations as listed below in object properties.	
Object Properties:		**Properties**
	intervalBefore:	Domain and Range: ProperInterval Inverse Property: *After*
	intervalMeets:	Domain and Range: ProperInterval Inverse Property: *MetBy*
	intervalOverlaps:	Domain and Range: ProperInterval Inverse Property: *OverlappedBy*
	intervalStarts:	Domain and Range: ProperInterval Inverse Property: *StartedBy*
	intervalDuring:	Domain and Range: ProperInterval Inverse Property: *Containts*
	intervalFinisches:	Domain and Range: ProperInterval

		Inverse Property: *FinischedBy*
	intervalEquals:	Domain and Range: ProperInterval Inverse Property: *Equals*

T4. OWL Sub-class	**DATE TIME INTERVAL**	
Entity definition:	This is a subclass of ProperInterval, defined using the multi-element DateTimeDescription	
Datatype Properties:	*xsdDateTime:*	Domain: DateTimeInterval Range: xsd:DateTime
Object Properties:		**Properties**
	hasDateTimeDuration:	Domain: DateTimeInterval Range: GeneralDateTimeDescription

T5. OWL Sub-class	**INSTANT**	
Entity definition:	This is a subclass of TemporalEntity which define a TemporalEntity with zero duration	
Datatype Properties:	*inXsdDateTime:*	Domain: Instant Range: xsd:DateTime
	Position of an instant, expressed using xsd:DateTime	
Object Properties:		**Properties**
	inTimePosition:	Domain: Instant Range: TimePosition
	inDateTime:	Domain: Instant Range: GeneralDateTimeDescription

Tab. 4 Specification of entities, relations and properties in the time ontology

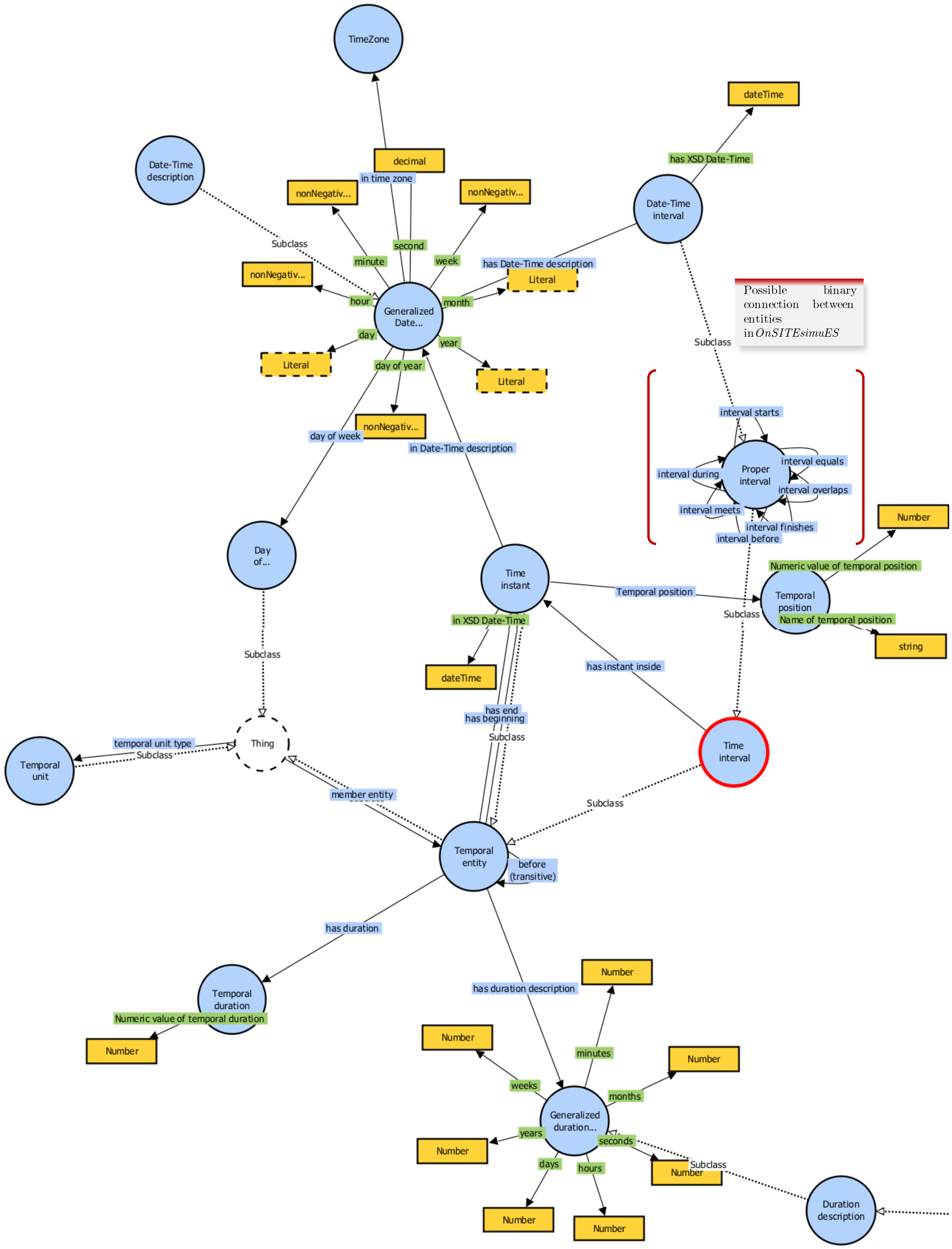


Figure 8-1 Time ontology edited in *Protégé* and visualized in a force-directed layout by means of made of visual elements. Blue circles represent the class hierarchy; blue rectangles represent relations, the green ones represent the property label of data-types, yellow rectangles represent the data assignment.

Chapter 9 Architecture of the rule-based reasoning machine

The decision to adopt a Knowledge-Base architecture based on ontologies to drive the construction site simulation is also justified by the fact that it is therefore possible to apply a reasoning machine (rule-based) in order to modify its internal structure adding new relationships or new data which allow the system to reach the generation of the *shortest construction completion sequence* of a given BIM. This achievement has been possible thanks to the development of a rule set which performs the core of the *Rule-based Reasoning Machine*[22] that uses rules to derive *conclusions* from *premises*.

The Semantic Web Rule Language (SWRL) has been used to express rules as well as logic. According Horrocks et al. (2004), it is an expressive OWL-based rule language allowing writing rules that can be expressed by using of all those OWL concepts and relationships before defined in the *Construction Process* (*Chapters 5,6,7,8*).

Even if the system, according to the scope of this *PhD thesis*, is equipped with a number of rules, furthermore by using this approach new ones may be configured and adjusted, resulting in a *flexible* and *extensible* system.

The *Rule Engine* obviously needs somewhere to store rules and to operate (Friedman, 2003). It shall be represented by the *rule base*, which contains all the rules the system knows. It is *Protégé*, the same ontology-based programming environment which has been used to edit the ontologies which contains all the pieces of information (*facts*) the rule engine will be working on.

As will be explained below, the rule engine has been integrated in the system architecture and the suggested interaction with the 'Construction Process Ontology' is presented on Figure 9-1 and explained step-by-step below:

(1) The Construction Process Ontology, edited in *Protégé*, contains a network (classes, relationships and properties) that underpins the system for the construction simulation. This ontological network is structured in several *facts* such as: 'A Construction Method requires some Micro-Level Workspaces', 'A resource handles an Activity' or 'A Labor Crew Space has a Reference Object' and so forth. Such ontological structure of the KB is here called *Working Memory,* due to the fact that it actively holds the information the rule

[22] In this research '*Rule-based Reasoning Machine'* and '*Rule Engine*' are used interchangeably.

engine will operate on (Fortineau et al., 2012).

(2) In parallel a Rule-Engine contains rules that are stored in the *Configuration Memory*, edited in *Protégé* just the same by using a predefined structure as beyond illustrated in Figure 9-2.

(3) Once the Reasoning Machine knows facts and rules to fire, it is activated and the configuration results are produced.

(4) As a last step, the reasoning machine adds, removes and modifies facts in the working memory (KB) and the updated *Construction Process Ontology* will be available. It should contain the 'earliest construction completion sequence' of the given BIM, the system was intended to generate.

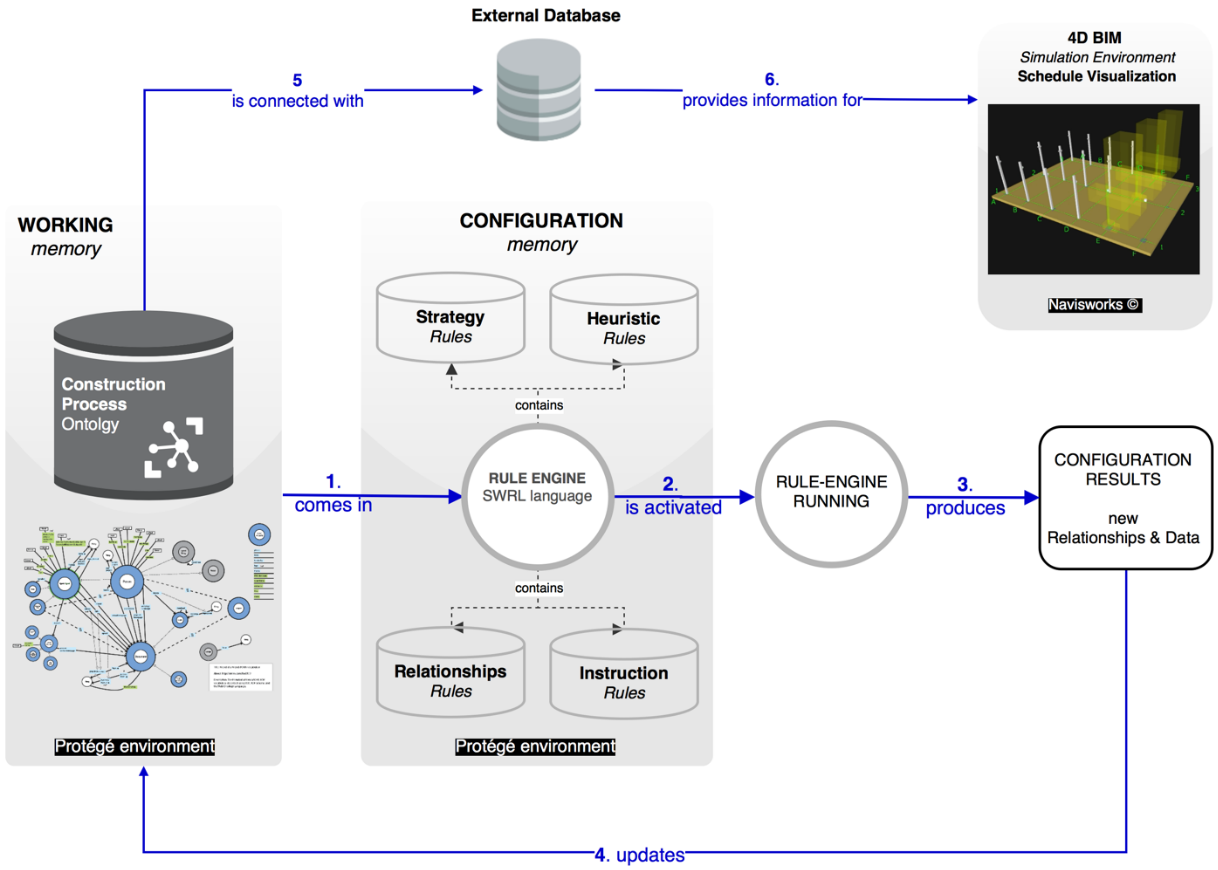


Figure 9-1 Graphical representation of the operating model of the rule-based reasoning machine with reference to the knowledge base.

9.1. The rule-set architecture

In the proposed system architecture, rules have been edited in SWRL language (Horrocks, 2004) and are implication rules. Hence, their syntax is of the following form (Muna and

Ramanib, 2011):

$$\textit{\textbf{Antecedent}} \rightarrow \textit{\textbf{Consequent}}$$

This syntax means that the conditions specified in the consequent must hold whenever the conditions specified in the antecedent are satisfied.

Both antecedent and consequent are conjunctions of atoms. *Atoms* can be of the form *OWL class* C(x), *OWL property* P(x,y), *sameAs*(x,y), *differentFrom*(x,y), and *SWRL built-in* functions, where x and y are OWL individuals or OWL data-values, the same edited in the ontologies definition.

Therefore, rules have been developed following the structure, before mentioned, which is graphically specified in figure 9-3. The graph depicts the formal components that could constitute each rule, their properties and relationships. One simple example, however, included in our rule-engine, is also presented. By using the same framework, each rule is presented with the same notation and description format (Tab. 5).

We have considered three main *types of rules* which allow the system to operate in order to reach different objectives for what it has been designed:

(a) ***Relationships*** **rules**:

These rules are used to generate new relationships between entities within the construction process ontology. They play a pivotal role in order to connect OWL individuals (e.g., resources, workspaces, constraints, activities, etc.) with the building products and allocate the same time duration and position to all those entities in a network every time that to only one entity is assigned a time duration or position.

(b) ***Instruction*** **rules**:

This rule set is responsible to manage the property set assigned to the Resources, due to the fact that a range of different resources with different site capabilities have been implemented. SWRL rules have been developed for Aggregate and Atomic Resource by now. Others may be designed in future developments of this PhD thesis.

(c) ***Strategy*** **rules**

This rule package, of major interest, is designed to manage the planning strategy that the system will have to adopt in order to generate the construction schedule. According to (Benevolenskiy, 2012) a strategy is a set of rules used for a special configuration goal. In our system, the goal is to generate the construction schedule with the minimum completion time by solving all the workspaces conflicts considering the fact that each building product is inextricably linked with all those workspaces which are demanded by its construction method. Five configuration strategies -rules- have been designed (Figure 9-1) and illustrated below. In the next paragraph the details of each strategy rule are presented by using a specific table.

(1) According to the 1st one, each building product is linked with the *workspaces* needed to be

built. This is the main assumption of this research. It is carried out by automatically assigning the same time duration and time position to both building products and related workspaces. (see rule 3.1)

(2) According to the 2nd strategy, by using the information extracted from the IFC data structure of the given BIM the *structural sequence* is simulated (column before beam, beam before slab and so forth). This rule doesn't consider workspaces due to the fact that the structural sequence is independent from the workspaces. (see rule 3.3)

(3) The 3rd strategy contains the *conflict resolution approach* by using the results of the workspaces conflicts checking process, exported from *Navisworks*© with (*.*txt*) file extension and entered in the rule-engine (see rule 3.2)

(4) The 4th strategy could be considered an extension of the third rule. The aforementioned 'conflict resolution approach' is carried out by using a *space overlapping ratio*. The conflicts, between workspaces of two different activities, with a spatial dimension below a minimum threshold to be assigned within the ontology are not considered to be a spatial constraint which means the activities can be overlapping. (see rule 3.4)

(5) According to the last strategy the building is constructed *bottom up* and *from left to right*. (see rule 3.5). It does not collide this the others and it is designed to validate the model in order to verify if the total amount of time required for the shortest schedule changes when a predefined construction progress rule is imposed.

9.2. SWRL-Rules specification

In this paragraph the planning rules which drive the schedule generation are specified. They have been designed according to the operational framework in Figure 9.2 and each one summarized in a specific table which contains the contents explained in Tab. 5.

(1) **Rule Name**	A name to each rule is assigned
(2) **Problem statement**	An informal rule-description is provided to describe its aim
(3) **Rule Creation**	Here a formal description of the rule is defined by using an IF-THEN structure and ontological entities included in the KB
(4) **Graphical representation**	A graphical representation of entities, relations and properties is provided highlighting new facts inferred by the rule-engine
(5) **Rule Translation**	The IF-THEN structure presented in step (3) is computerized by using a script in SWRL language

Tab. 5 Process of designing rules and description of main steps that will be presented in the rules tabs

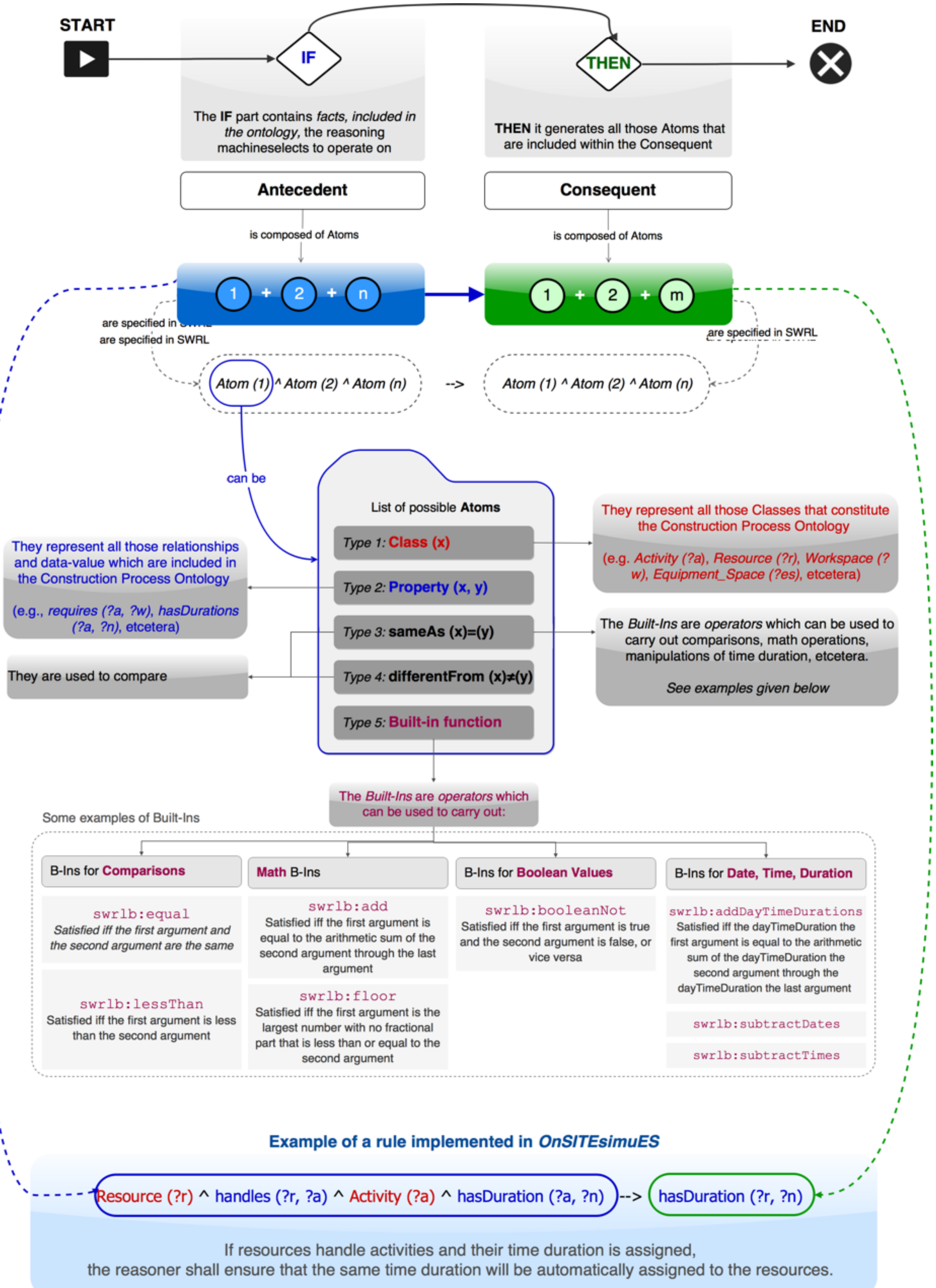


Figure 9-2 Operating diagram which depicts the rule structure and a list of operators able to redefine facts (relations and properties) of the knowledge base by means of the rule engine

Allocation of Time Duration to the Resources

Step 1 — **Problem statement**

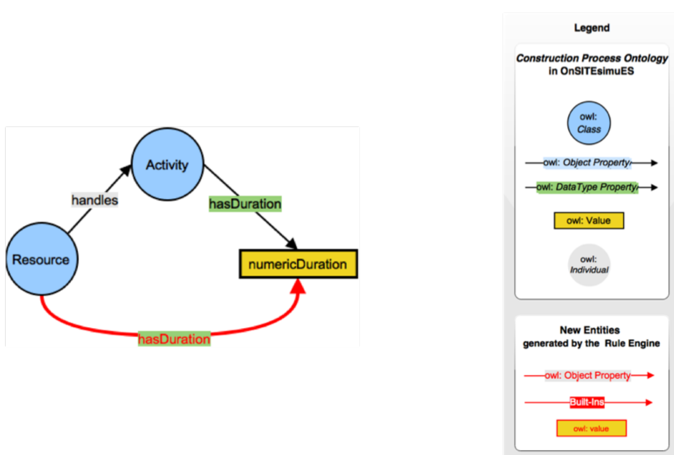
When the user or the system assign a time duration to an activity and some resources handle the activity itself, the reasoner shall ensure that the same time duration will be automatically assigned to the resources.

Step 2 — **Rule Creation**

3.1 Informal Representation — **Family Rule** — Relationships (1.1)

	RELATIONSHIP SCENARIO 1		
Antecedent	**IF**	<condition 1>	<Resource *handles* Activity>
	AND	<condition 2>	<Activity *hasDuration* Numeric Duration :[x]>
Consequent	**THEN**	<relation 1>	<Resource *hasDuration* Numeric Duration :[x]>
	END RELATIONSHIP		

3.2 Graphical representation of the scenario



Step 3 — **Rule translation in SWRL**

Resource (?r) ^ handles (?r, ?a) ^ Activity (?a) ^ hasDuration (?a, ?n ha) -> Duration (?r, ?n)

Allocation of Time Duration to the Workspaces

Step 1 — **Problem statement**

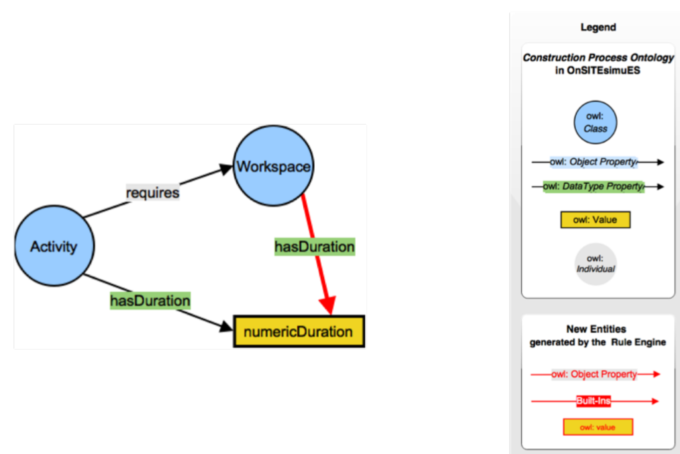
This rule assigns a time duration to the individuals that represents the workspaces within the ontology. In the event that an Activity is linked with a Data-Type property which describes its time duration and at the same time it requires some workspaces, the reasoner will assign the same time duration to the workspaces themselves.

Step 3 — **Rule Creation**

3.1 Informal Representation — **Family Rule** — Relationships (1.2)

	RELATIONSHIP SCENARIO 2		
Antecedent	**IF**	<condition 1>	<Activity *requires* Workspace>
	AND	<condition 2>	<Activity *hasDuration* Numeric Duration :[x]>
Consequent	**THEN**	<relation 1>	<Workspace *hasDuration* Numeric Duration :[x]>
	END RELATIONSHIP		

3.2 Graphical representation of the scenario



Step 5 — **Rule translation in SWRL**

Activity(?a) ^ requires (?a, ?ws) ^hasDuration n(?a, ?n) -> WorkSpace (?ws) ^ hasDuration (?ws, ?n)

Activities: Assignment of Time Duration

Step 1 — **Problem statement**

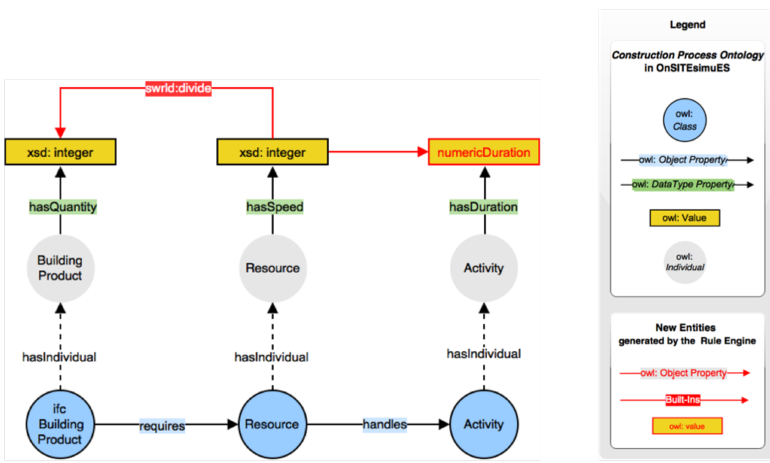
This rule calculates the time duration which the reasoner assigns to each activity when the 'forward speed' (*Data-type property*) of the resources, that handle the activity itself, has been assigned.

Step 3 — **Rule Creation**

3.1 Informal Representation — **Family Rule** — Relationships (1.3)

	RELATIONSHIP 1.2		
Antecedent	**IF**	<condition 1>	<ifcBuildingProduct *requires* Resource>
	AND	<condition 2>	<Resource *handles* Activity>
	AND	<condition 3>	<Activity *handles* Activity>
Consequent	**THEN**	<relation 1>	<Activity *hasDuration* Numeric Duration :[x]>
	END RELATIONSHIP		

3.2 Graphical representation of the scenario



Step 5 — **Rule translation in SWRL**

ifcBuildingProduct (?ifc) ^ requires (?i,?r) ^ hasQuantity (?ifc,?q) ^ Resource (?r) ^ handles (?r,?a) ^ hasSpeed (?r,?s) ^ swrld:divide (?x,?s,?q) -> hasDuration (?a,?x)

Phases and Processes: Assignment of Time Duration

Step 1 — **Problem statement**

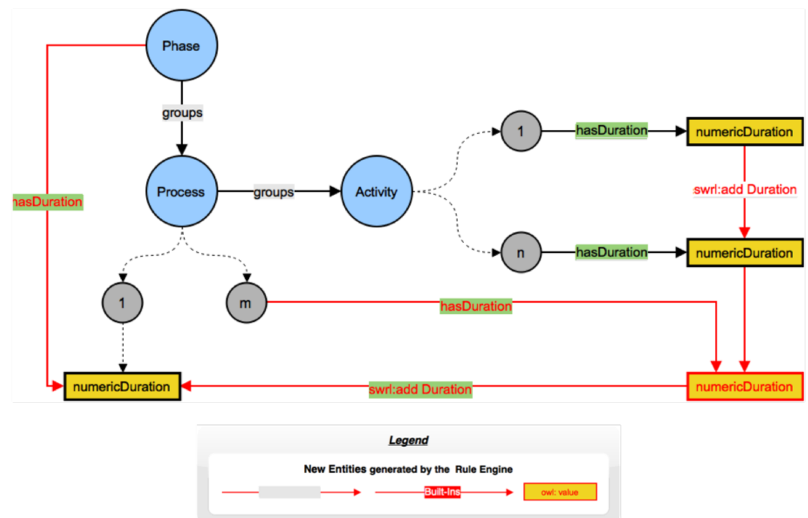
This rule calculates the time duration which the reasoner automatically assigns to the Processes and Phases according to the specification included in the Construction Scheduling Ontology.

Step 3 — **Rule Creation**

3.1 Informal Representation — **Family Rule** — Relationships (1.4)

	RELATIONSHIP 1.2		
Antecedent	**IF**	<condition 1>	<Activity *hasDuration* Numeric Duration :[x]>
	AND	<condition 2>	<Activity *intervalMeets* Activity>
	AND	<condition 3>	<Activity *intervalOverlaps* Activity>
Consequent	**THEN**	<relation 1>	<Process *hasDuration* Numeric Duration :[x]>
	AND	<condition 4>	<Process *intervalMeets* Process>
	AND	<condition 5>	<Process *intervalOverlaps* Process>
	THEN	<relation 2>	<Phase *hasDuration* Numeric Duration :[x]>
	END RELATIONSHIP		

3.2 Graphical representation of the scenario



Step 5 — **Rule translation in SWRL**

Activity (?a) ^ haDuration (?a,?x) ^ intervalMeets (?a, ?a) ^ intervalOverlaps (?a, ?a) ^ swrlb:addDayTimeDurations (?a,?a,?t1) -> Process (?p) ^ hasDuration (?p,?t1) ^ intervalMeets (?p, ?p) ^ intervalOverlaps (?p, ?p) ^ swrlb:addDayTimeDurations (?p,?p,?t2) -> Phase (?ph) ^ hasDuration (?ph,?t2)

Adding construction process information to the ifcBuildingProduct

Step 1	Problem statement

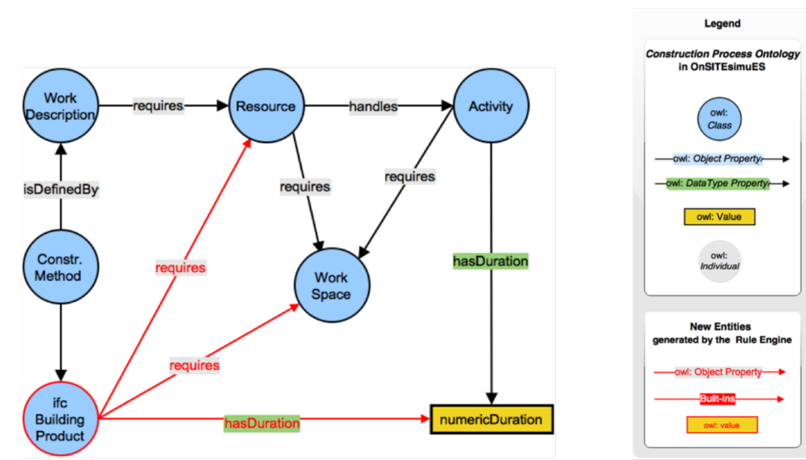
By using this rule each building product has been linked with the workspaces and resources and time duration of the activities that its construction method requires. This approach generates a network of relationships and properties which handles the system to be reconfigured automatically if changes occur.

Step 3	Rule Creation

3.1 Informal Representation — **Family Rule** — Strategy 1

Antecedent	**INSTRUCTION** SCENARIO 1		
	IF	<condition 1>	ConstructionMethod *isDefinedBy* WorkDescription
	AND	<condition 2>	WorkDescription *defines* Demand
	AND	<condition 3>	Demand *requires* Resource
	AND	<condition 4>	Resource *handles* Activity
	AND	<condition 5>	Activity *requires* Workspace
Consequent	**THEN**	<relation 1>	ConstructionMethod *requires* Workspace
	END INSTRUCTION		

3.2 Graphical representation of the scenario



Step 5	Rule translation in SWRL

ifcBuildingProduct (?ibp) ^ Construction Method (?cm) ^ isDefinedBy (?c, ?wd) ^ Work Description (?wd) ^ defines (?wd, ?d) ^ Demand (?d) ^ requires (?d, ?r) ^ Resource (?r) ^ handles (?r, ?a) ^ Activity (?a) ^ requires (?a, ?ws) ^ hasDuration (?a,?t1) → requires (?cm, ?ws) ^ requires (?ibp, ?r) ^ requires (?ibp, ?ws) ^ hasDuration (ibp?, ?t1)

Spatial conflict resolution

Step 1	Problem statement

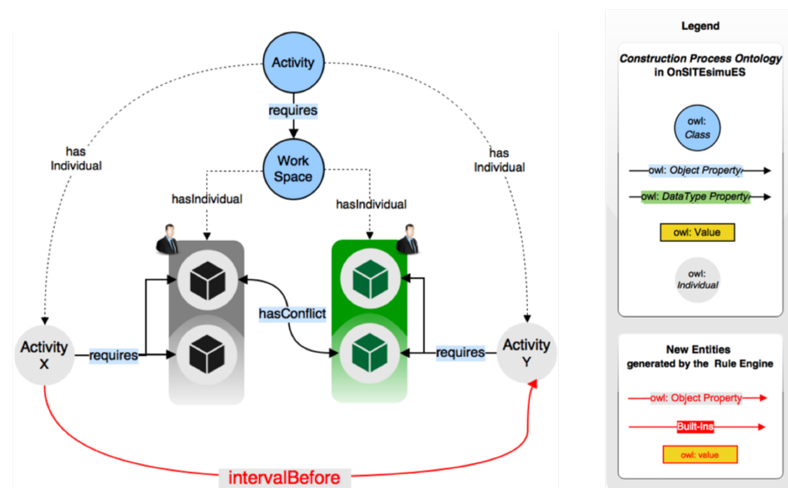
This rule is designed to solve the spatial conflict between two workspaces. By means of data included in the 'Clash Report' a spatial relation is added in the ontology -by using an OWL-relation (*hasInteraction*)- between the conflicting workspaces. Then the reasoner generates a new time relation (i.e., *IntervalBefore)* between the activities handled by the some workspaces.

Step 3	Rule Creation

3.1 Informal Representation — **Family Rule** — Strategy 2

Antecedent	**INSTRUCTION** SCENARIO 2		
	IF	<condition 1>	Activity *requires* WorkSpace
	AND	<condition 2>	Workspace *hasInteraction* Workspace
Consequent	**THEN**	<relation 1>	Activity *intervalBefore* Activity
	END INSTRUCTION		

3.2 Graphical representation of the scenario



Step 5	Rule translation in SWRL

Activity(?a) ^ WorkSpace(?w) ^ requires(?a,?ws) ^ hasInteraction(?ws,?ws) -> IntervalBefore(?a, ?ws)

Conflict resolution rule by using a space overlapping ratio

Step 1 — **Problem statement**

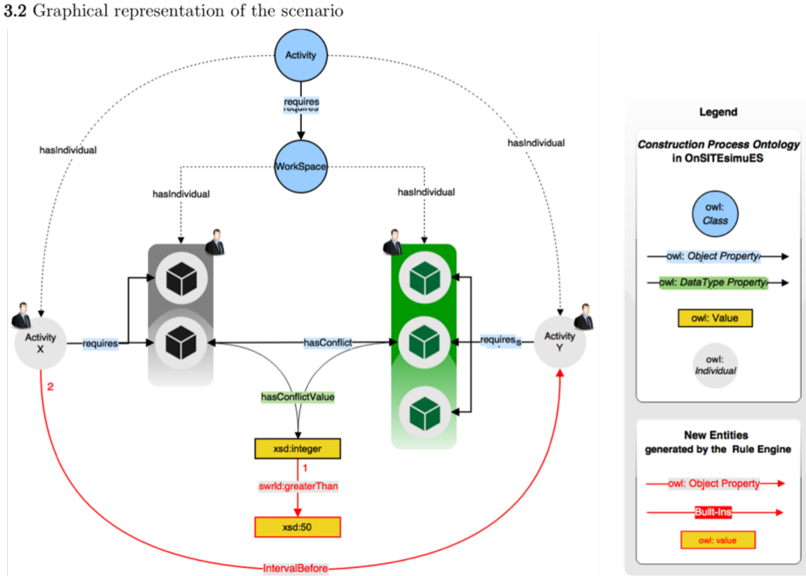
The strategy could be considered an extension of the strategy 2. The aforementioned 'conflict resolution approach' is carried out by using a space overlapping ratio. The conflicts, between workspaces of two different activities, with a spatial dimension below a minimum threshold to be assigned (in the test case it is 50 cm) within the ontology are not considered to be a spatial constraint which means the activities can be overlapping.

Step 3 — **Rule Creation**

3.1 Informal Representation — **Family Rule** — Strategy 3

Antecedent	**INSTRUCTION** STRATEGY 3		
	IF	<condition 1>	Activity *requires* WorkSpace
	AND	<condition 2>	Workspace *hasConflict* Workspace
	AND	<condition 3>	Workspace *hasConflictValue* greaterThan:[50]
Consequent	**THEN**	<relation 1>	Activity *intervalBefore* Activity
	END INSTRUCTION		

3.2 Graphical representation of the scenario



Step 5 — **Rule translation in SWRL**

Activity(?a) ^ WorkSpace(?w) ^ requires(?a,?ws) ^ hasConflict (?ws,?ws) ^ hasConflictValue (?ws,?cv) ^ swrb:greaterThan (?cv, 50) -> IntervalBefore(?a, ?ws)

Management of a construction sequence: *bottom-up* and *left-to-right*

Problem statement

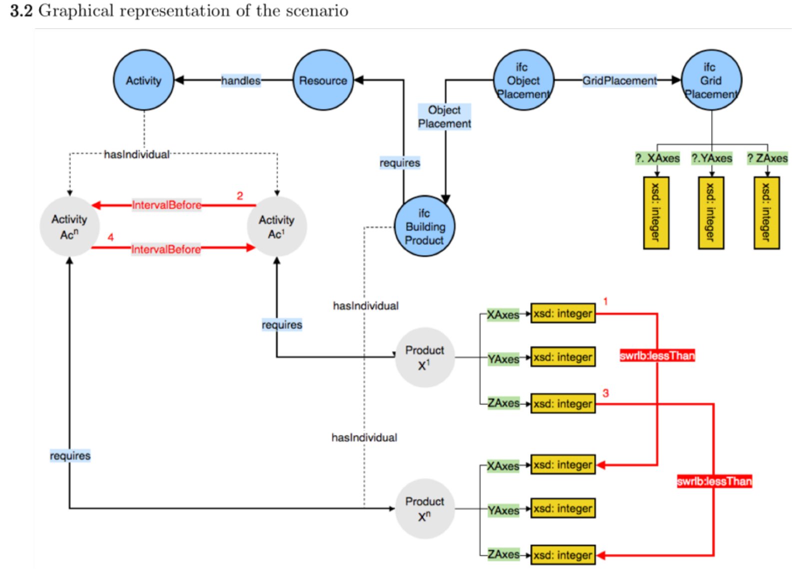
According to this strategy the building is constructed *bottom up* and from *left to right*. The sequence is reached by using a pairwise comparison between the coordinate system of the building products within the given BIM. For example, to ensure the left-to-right construction direction, in the comparison the object which has a placement location with a lover x-axis value is constructed before. A Built-In for comparison (*swrlb:lessThan*) has been used to make comparisons by using the reasoner.

Rule Creation

3.1 Informal Representation — **Family Rule** — Strategy 4

INSTRUCTION STRATEGY 3		
IF	<condition 1>	ifcBuildingProduct *objectPlaceme\nt* X-Axes
AND	<condition 2>	ifcBuildingProduct *requires* Activity
AND	<condition 3>	X-Axes lessThan X-Axes
THEN	<relation 1>	Activity *intervalBefore* Activity
AND	<condition 4>	Z-axes lessThan Z-axes
THEN	<relation 2>	Activity *intervalBefore* Activity
END INSTRUCTION		

3.2 Graphical representation of the scenario



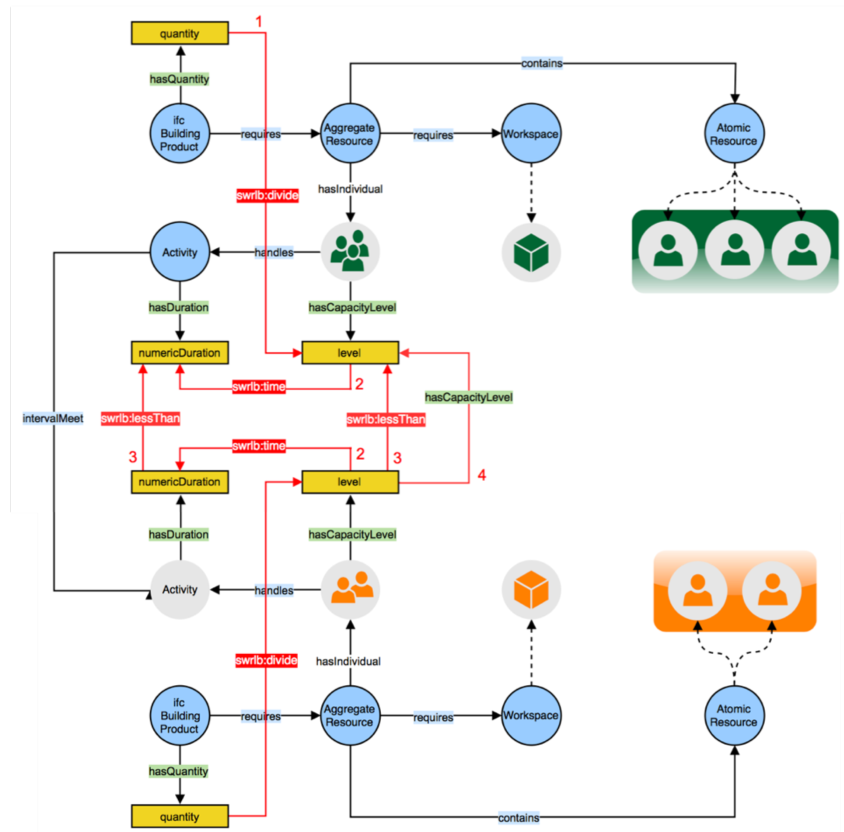
Rule translation in SWRL

IfcBuildingProduct (?ibp) ^ Activity (?a) ^ requires (?ibp, ?a) ^ objectPlacement (?ibp, ?iop) ^ ifcGridPlacement (?iop,?x,?y,?z) ^ swrlb:lessThan(?x, ?x) -> IntervalBefore(?a, ?a)

Balancing Aggregate Resources by using Atomic Resources	
Step **1**	**Problem statement**

According to this strategy if two activities run in the same time period because of no conflicts have been detected between them but they have different time duration due the different '*CapacityLevel*' of the resources which handles the activities themselves, the reasoner will ensure the to balance the '*CapacityLevel*' in order to ensure a shorter time duration.

3.2 Graphical representation of the scenario



Step **5**	**Rule translation in SWRL**

Activity (?a1) ^ hasDuration (?a, ?t1) ^ Resource (?r) ^ AtomicResource (?ar) ^ handles (?r, ?a1) ^ contains (?r, ?ar) ^ hasCapacityLevel (c1) ^ IntervalMeets (?a1, ?a) ^ swrlb:lessThan (?t1, ?t) ->

Chapter 10 Built-in algorithms for workspaces management

Workspaces planning and modelling are two crucial points in this research. Having stored in the Knowledge-Base their formal representation and topological relationships in the form of ontologies, the workspaces management process is performed by using two built-in algorithms followed by the workspaces conflicts checking process (Figure 10-1):

(1) built-in algorithm to generate the workspaces spatial allocation patterns;
(2) built-in algorithm to automatically sculpt workspaces geometries.

Their own operating mechanisms are sequential in the sense that the output data of the first one triggers the operation of the other as depicted in the figure below.

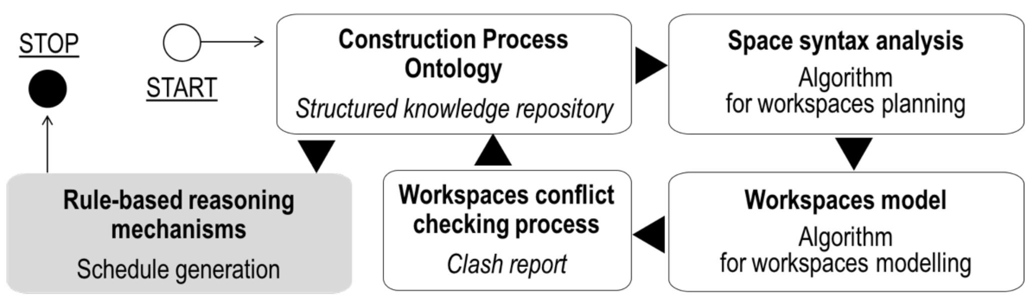


Figure 10-1 Working steps performed by the system for the workspace management process

10.1. Algorithm for workspaces planning

The whole design workflow, proposed in this model, is about going from an abstract graph description of workspaces topological structure and their interactions -included in the ontology- to find their optimal planar allocation (constraint-based) with reference to the building objects those spaces are referred. This analyzing that graph and finding the feasible geometric configuration as well as layout allocation that admits the proposed graph of connections.

The computerization of such a concept has been possible with a built-in algorithm able to manage a parametric design workflow developed *Grasshopper*©, (a graphical algorithm editor tightly integrated with a 3-D modeling environment).

This digital programming environment for parametric algorithm contains many operators to perform mathematical operations, logical operations, evaluate conditions and manipulate sets of data. The algorithm is designed by combining such operators in a structured sequence able to manage our planning purposes. In Figure 10-2 some operators are depicted.

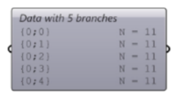
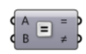
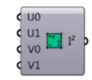
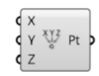
Category	Function	Description	Operator
Primitive	File Path	Contains a collection of file paths	
Geometry	Circle Parameter	Represents a collection of Circle primitives	
Utilities	Viewer	A viewer for data structures	
Maths	Equality	Test for (in)equality of two numbers	
Domain	Construct Domain[2]	Create a two-dimensional domain from four numbers	
Sets	Random	Represents a collection of Circle primitives	
Point	Viewer	A viewer for data structures	

Figure 10-2 Some graphical operators able to perform different kind of operations and data manipulations in Grasshopper's editing environment

In the proposed system architecture, the algorithm presented in Nourian et al. (2013) has been extended and customized for our specific spatial planning purposes. Its main operating steps are below presented with reference to both algorithm operators -named (*op. X*)- and their graphical outputs in Figure 10-3.

i. Workspaces graph representation

As the starting point, a number of randomly located points, as for depicting the center of workspaces, are generated in a CAD environment by using a first operator (*op.1*) (Figure 10-3-B). At the same time, a corresponding list of dimension values (`has_Lenght`, `has_Width`) and identification numbers (`has_ID`) are imported from the ontology and a set of operators, first assign colors to the workspaces to make them more recognizable and then generate one circle around each workspaces' center points. Their sizes are deduced by the rectangular areas of workspaces as suggested by the user in the ontology; they are equal (*op.2*).

In this way, a first *workspaces map representation* is generated. This is carried out for each construction methods included in the scheduling ontology as shown in Figure 10-3-C in the case of Column Installation in anticipation of the case study.

ii. Workspaces connectivity graph

Subsequently, the representation of relationships between workspaces plays a fundamental role for the spatial analyzes. In this regard, an operator -linked with the previous- establishes connections between every pair of points (circles representing workspaces) that have a topological interaction according to their ontological structure. This is visualized with red connection lines (Figure 10-3-D). In the Nourian's algorithm, this was manually carried out instead of our extension where this has been automated.

The connectivity requirements are the topological interaction values (`has_Topological_Interaction`) again managed in the ontology that can be seen as an adjacency requirements set included in the algorithm itself (*op.3*).

iii. Space syntax analysis

Having generated the connectivity graph, in theory of *space syntax analysis*[23] there would be a number of syntactic measures to be calculated (e.g., connectivity, depth, control value, local and global integration) (Sayed et al. 2014). In our system architecture, we have considered the '*depth*' defined as the smallest number of syntactic steps (in topological meaning) that are needed to reach one space from another.

On this basis, going ahead in the algorithm workflow, an operator generates a theoretical distance measured between two workspaces in the connectivity graph and automatically visualizes such distances on depth levels by using a 'justified plan graph' (*op.4*) (Figure 10-3-E).

Then it re-distributes the workspaces on the depth levels in keeping with the connectivity map in the previous step. Depending on the number of workspaces that each construction methods contains -e.g. five spaces for the column installation- one configuration for each space by using depth levels is generated. It represents the point of view of a laborer getting in position in that workspace on '*depth-0*'.

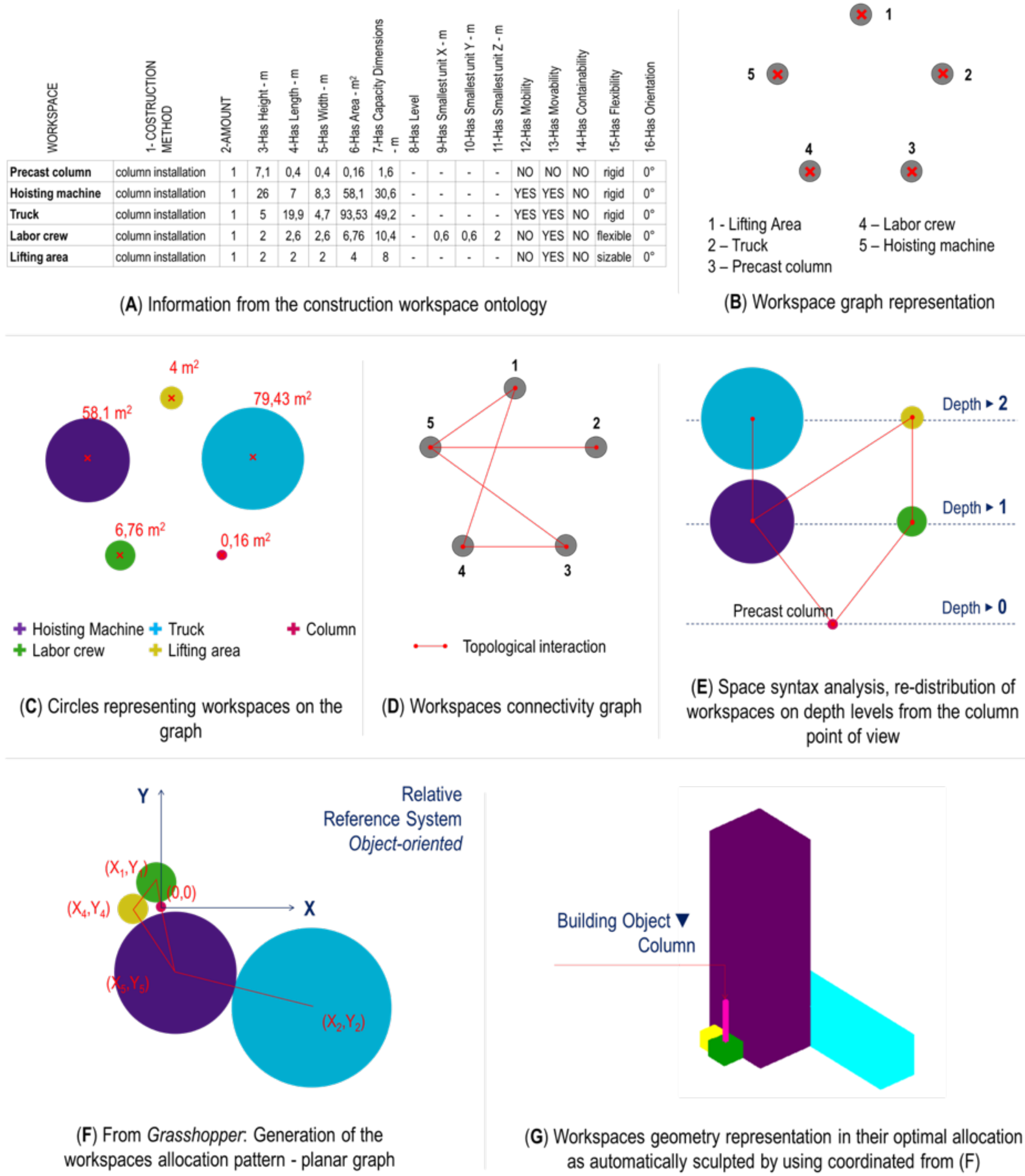
iv. Generation of the workspaces allocation pattern

Once the system is provided with workspaces connectivity values as graphically depicted in the related connectivity map, the tool contains a force-directed graph-drawing operator (*op. 5*) which is able to generate a *bubble diagram* representing the optimal *workspaces allocation pattern* based on user-constraints before setted in the ontology.

This function works translating the interaction values between workspaces (`has_Interaction_Value`) in a set of *attractive* and *repulsive forces.* The forces act recursively on the graph vertices, seek a 'relax' situation for a graph, and provide the system with a graphical representation of the given solution (Fig. 10-3-F). The output is a bubble diagram – one for each construction method included in the system architecture- compliant with the specified workspaces dimensions and the connectivity values.

---

[23] *Space syntax* theory has been used to understand environmental behavior, cognition, and evidence-based design phenomena in existing healthcare settings, such as wayfinding in hospitals, nurse movement behavior in surgical units, privacy preferences in wards, and many more (Haq, Luo, 2012). However, no studies attempted to utilize space syntax theory or techniques in the workspaces planning for construction scheduling. This is a novel aim for this study that explore such an implementation of space adjacency analysis to allocate workspaces in site as well as its integration in a more structured expert system.



WORKSPACE	1- COSTRUCTION METHOD	2-AMOUNT	3-Has Height - m	4-Has Length - m	5-Has Width - m	6-Has Area - m²	7-Has Capacity Dimensions - m	8-Has Level	9-Has Smallest unit X - m	10-Has Smallest unit Y - m	11-Has Smallest unit Z - m	12-Has Mobility	13-Has Movability	14-Has Containability	15-Has Flexibility	16-Has Orientation
Precast column	column installation	1	7,1	0,4	0,4	0,16	1,6	-	-	-	-	NO	NO	NO	rigid	0°
Hoisting machine	column installation	1	26	7	8,3	58,1	30,6	-	-	-	-	YES	YES	NO	rigid	0°
Truck	column installation	1	5	19,9	4,7	93,53	49,2	-	-	-	-	YES	YES	NO	rigid	0°
Labor crew	column installation	1	2	2,6	2,6	6,76	10,4	-	0,6	0,6	2	NO	YES	NO	flexible	0°
Lifting area	column installation	1	2	2	2	4	8	-	-	-	-	NO	YES	NO	sizable	0°

Figure 10-3 Graphical outputs from (A) to (G) obtained by using the built-in algorithm to define the workspaces configuration pattern of a construction method. In the figure are presented a preview of the outputs for the case of the column installation which is one the five methods included in the case study

v. Workspaces allocations: spatial coordinates

Once a workspaces configuration pattern is deducted, it remains to extract the spatial allocation coordinates -on the X-axis and Y-axis- from the generated bubble diagram due to the fact that the model considers those workspaces as located at the same height (Z-axis) of their connected building object. This is crucial to model the workspaces geometries in a BIM environment able to produce the workspace conflicts checking process. One final operator (*op.6*) gets into this workflow extracting those coordinates values on an object-oriented reference system -centered in the building object-. The configuration pattern is so applied to all the building objects ontologically linked with the same construction method (Figure 10-4).

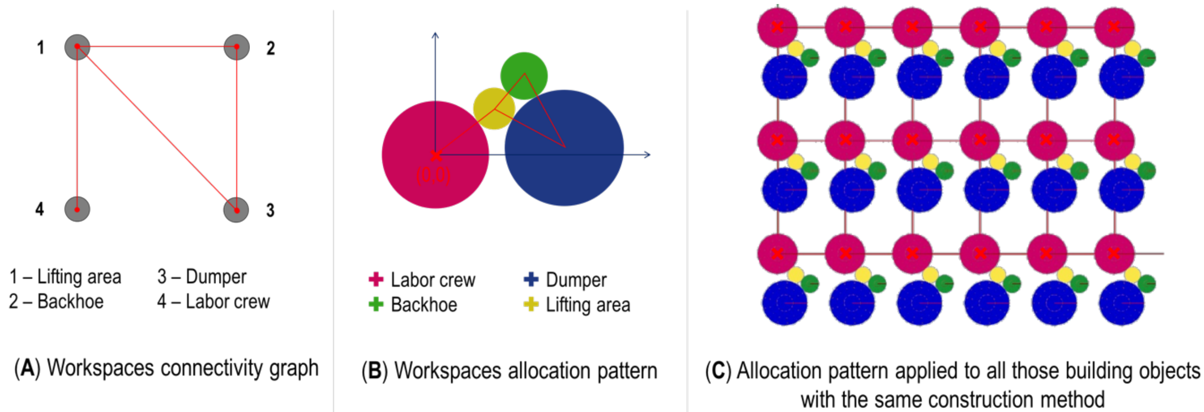


Figure 10-4 Application of workspaces configuration pattern to a number of building objects by using the built-in algorithm in Grasshopper environment. Construction method of plinth installation.

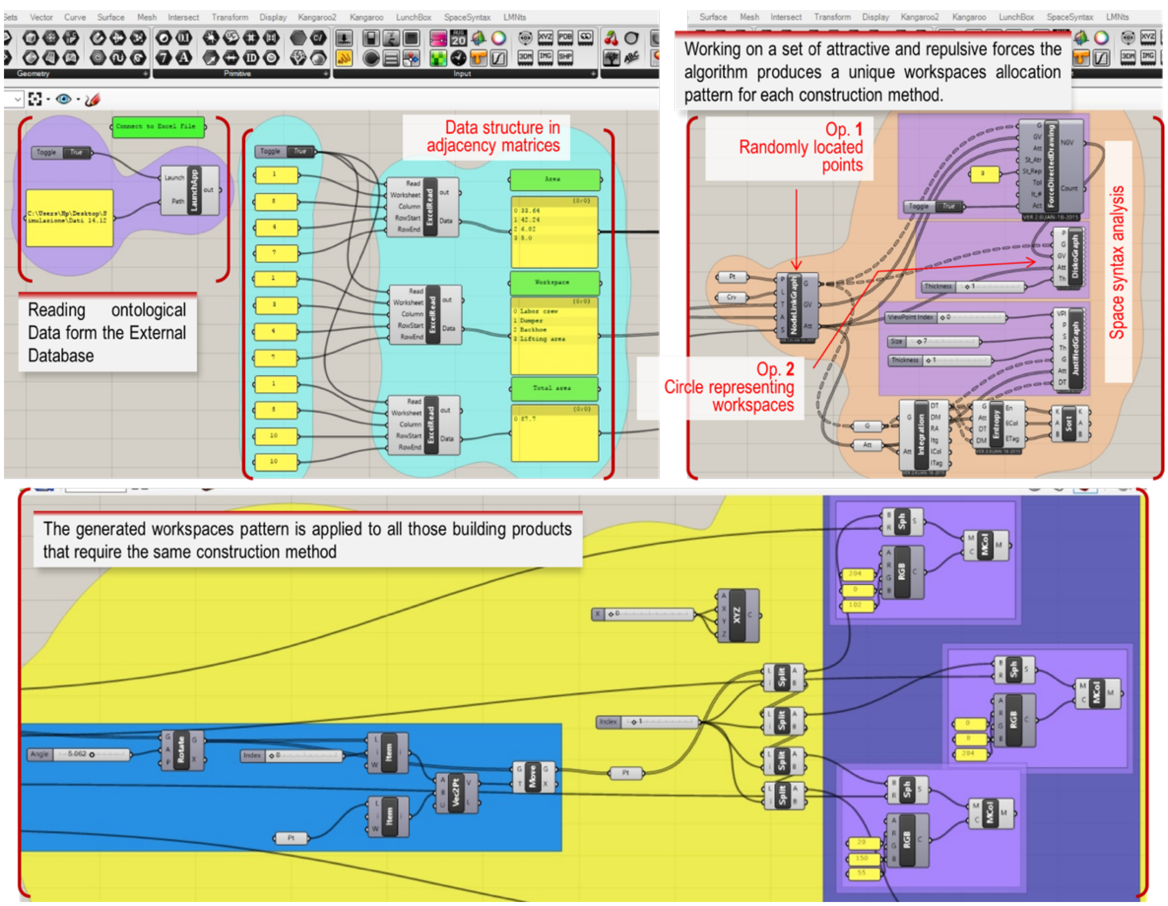


Figure 10-5 Algorithm with operators to find the workspaces configuration edited in a *Grasshopper's script*

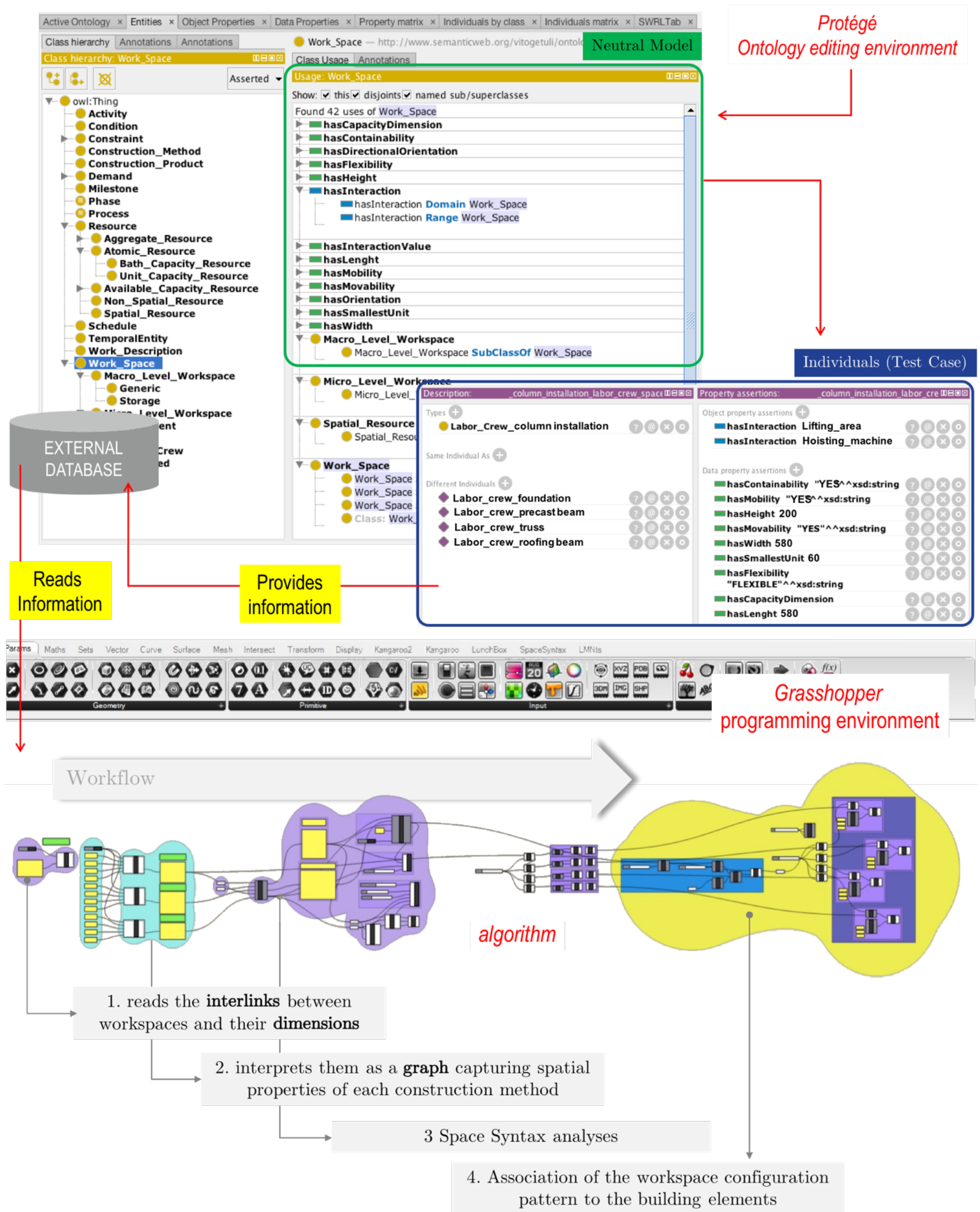


Figure 10-6 Snapshot of the entire algorithm from the programming environment (*Grasshopper's script*). The bubbles of different colors which group a number of operators depict the working steps to reach the workspaces configuration pattern of one construction method

An external data-base works as bridge, to transpose information, between the knowledge base (*Protégé*) and the algorithm committed to develop the Space Syntax Analysis (*Grasshopper*).

Chapter 11 Validation test and case study

For the validation of the Expert System, a simplified Building Information Model (BIM) of an industrial building has been considered, taking into account its structural subsystem but leaving aside the building finishing. According to the system architecture, results of the seven operational modules that perform different tasks and drive the system in bringing about a solution to the issue of construction schedule generation are presented below and graphically depicted in Figure 11-1.

Module 1. **Data collection and Knowledge-Base (KB) structuring**

In this step the pre-loaded conceptual data model –*construction process ontology*- has been specified adding 'individuals' within the ontology editing environment (*protégé*) both from the given BIM and the external user specially simulated. The ontology has been filled up primarily by the building objects information (98 items) and construction methods properties (6 methods). Doing so, the Knowledge Base has operated as structured data source for the next modules, especially when activating on it the rule-based reasoning mechanisms.

Module 2. **Generation of the 'Bounding-box Model'**

By using the information stored in the KB, the so-called 'Bounding-box model' has been generated. It is composed of 96 bounding boxes that surround the shape of 3D elements of each building objects.

Module 3. **Schedule generation: validation of the Structural construction sequence**

On the basis of data provided by the IFC to the knowledge-base, the system operated by means of the Rule Engine and the structural construction sequence is generated in terms of time relations between ontological entities. The most representative shots, needed to carry out a visual validation of the structural sequence have been extracted from the 4D BIM-based simulation ontology-driven as shown in Figure 11-9-10. The system generates a correct sequence (i.e., plinths before columns, columns before beams, beams before roofing beams, etc.) therefore it has been considered validated.

Module 4. **Workspaces planning process**

As anticipated, the given BIM is composed of six objects types therefore six corresponding construction methods have been specified. In order to test the working mechanism of the built-in planning algorithm, aiming to reach a spatial configuration pattern of each method, six summary sheets (Figure 11-11-12-13-14-15) set out the obtained results in terms of space syntax analyses and workspace allocation patterns.

The results show that the suggested workspaces configuration pattern strictly depends on the connectivity graph which in turn saves the workspaces interaction values, infact, even if three different construction methods (column, beam and roofing beam installation) require five spaces each with similar dimensions, their configuration patterns are significantly different. Therefore, the planning process has been considered validated.

Module 5. **Generation of the 'Full Bounding-Box Model**

By using the output-data coming from the previous module (planar coordinates of workspaces allocation on the X-axis and Y-axis with reference to the building objects), the 'bounding box model' before presented in module 2 has been enriched sculpting automatically the workspaces geometries around each building objects obtaining six hundred and eleven workspaces –located in their optimal configuration- required to the construction process.

Module 6. **Workspaces conflict checking process**

Therefore, the 'full bounding-box model' has undergone the workspaces conflicts checking process. One hundred and eighty conflicts have been detected and the clash report has been extracted in (*.*txt*) extension, enabling the rule-engine to solve those conflicts.

Module 7. **Schedule generation: validation of the 'earliest construction completion sequence'**

Finally, the rules execution phase extends the structural schedule (module 3) converting each conflict in new temporal relations and temporally reallocating the resources providing the system with the earliest construction completion sequence. Once again the suggested construction schedule has been visualized by means of a 4D simulation. The most representative shots, where it has been possible to validate the conflict resolution approach, have been extracted and presented in Figure 11-21-22-23-24).

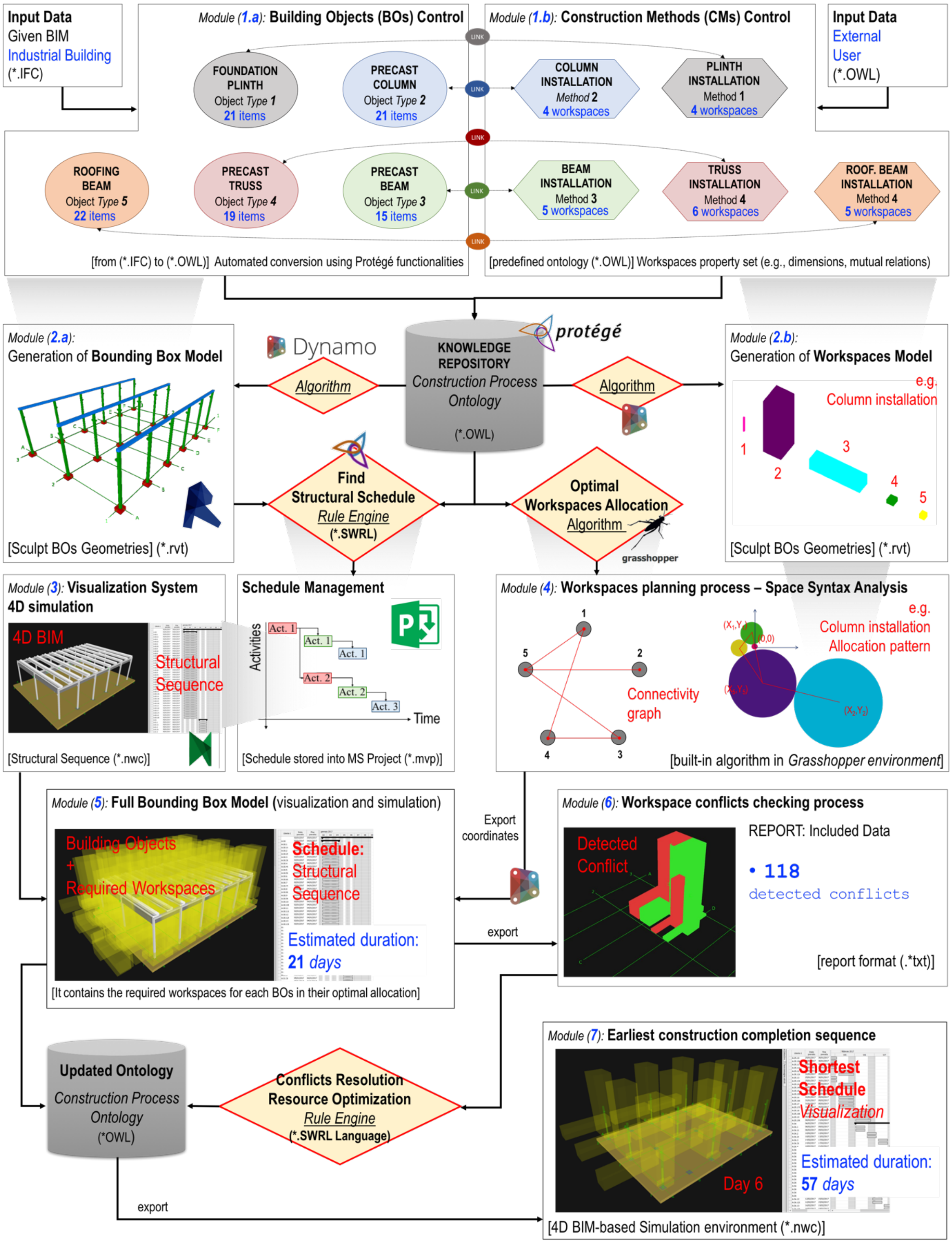


Figure 11-1 Data Management Protocol, application to a case study. The figure shows the operational modules of the proposed model. The generation and visualization of the earliest construction completion sequence of the building BIM was made possible and the model has been validated.

11.1 Module 1: Data collection and Knowledge-Base structuring

In the picture below in the first step an external user gets the required information in the predefined ontology by using *Protégé* capabilities to add individuals in terms of construction methods requirements.

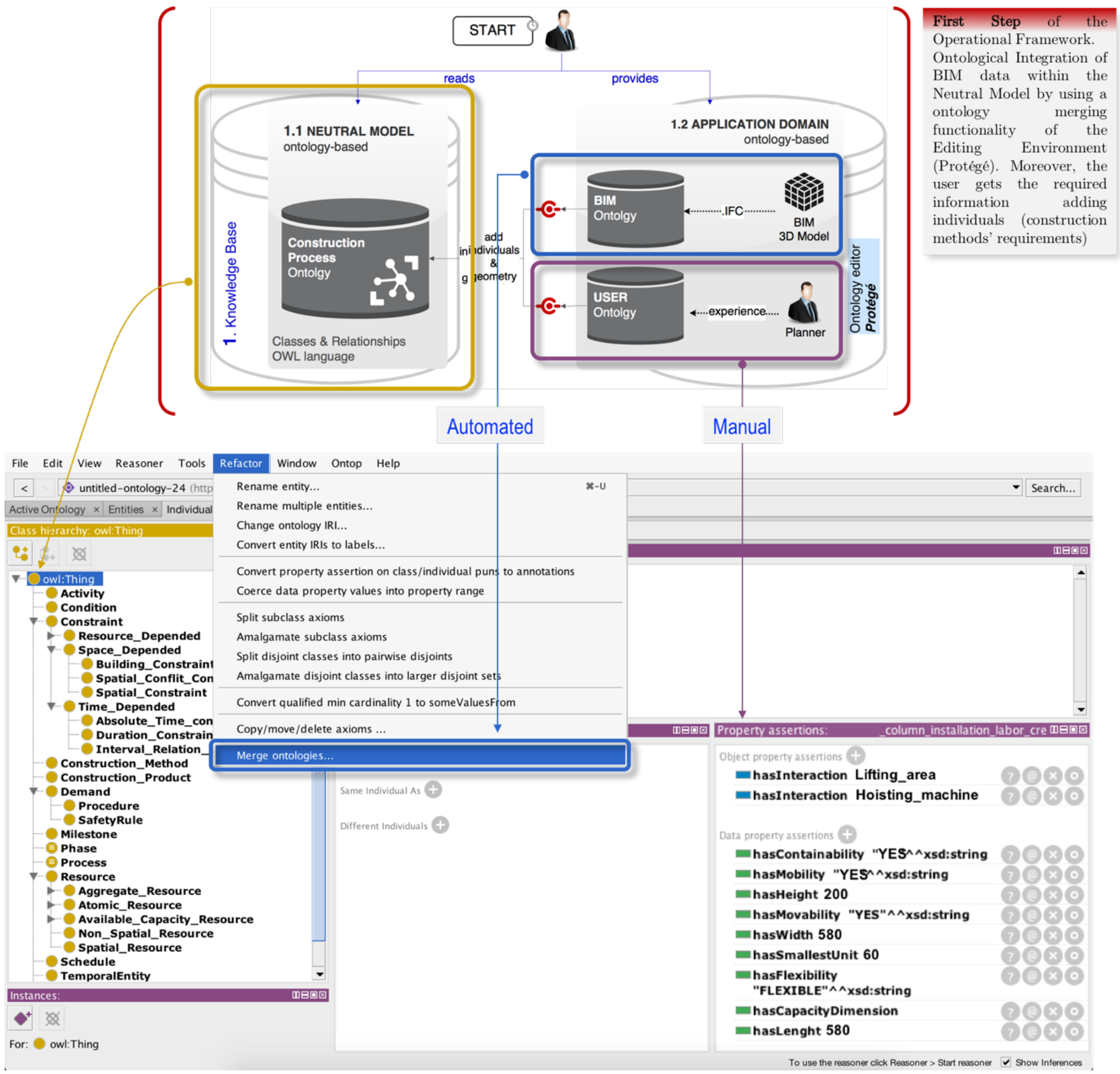


Figure 11-2 Construction Process Ontology within the ontology Editor Protégé. This working environment is used to merge the neutral model with the ontological information of the given BIM and user's experience

In order to introduce in the ontology, the BIM-model information, an IFC-to-OWL conversion process has been carried out and the IFC instances have been integrated within the knowledge base by using the merging capability of the editing environment (Figure 11-2). The building model is composed of five building objects types: (1) *Foundation Plinth* in the number of eighteen items; (2) *Precast Column* in the number of eighteen; (3) *Precast Beam*, fifteen products; (4) *Truss* in the number of nineteen items and (5) *Roofing Beam*, twenty-two elements.

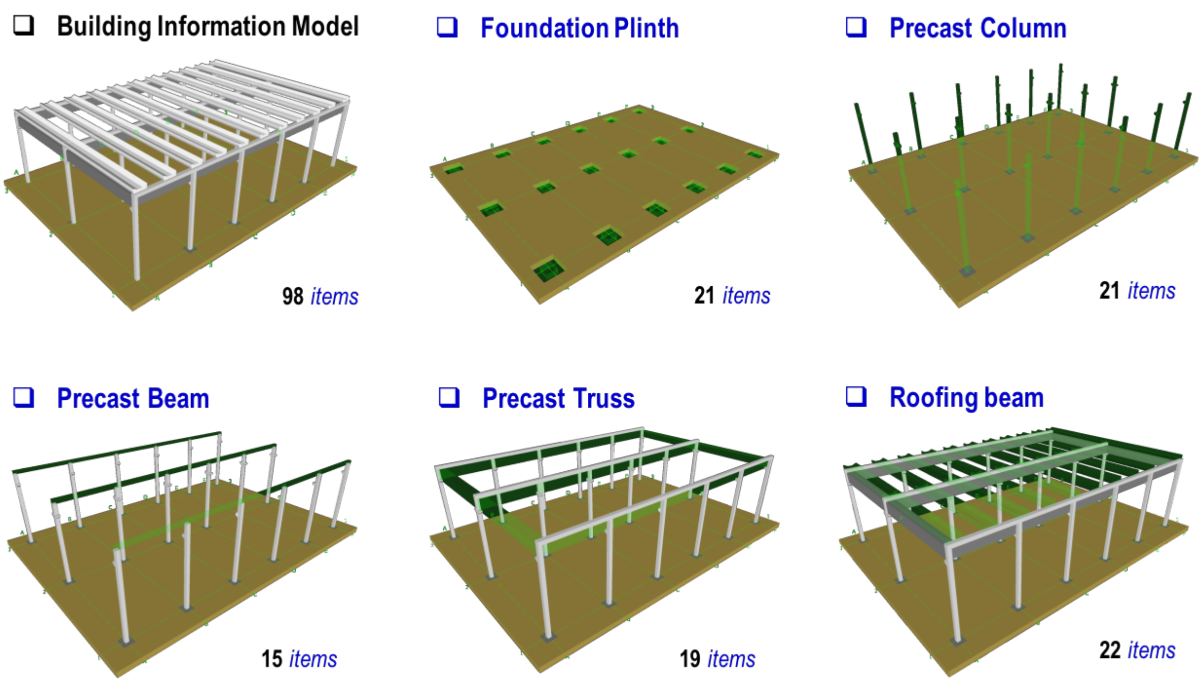


Figure 11-3 Building objects visualization (case study). Each figure groups a specific building object type.

Then, simulating what an external user would have to do in filling in the ontology the required information, each construction method has been specified:

1. Foundation plinth installation: details in Figure 11-4;
2. Precast column installation: details in Figure 11-4;
3. Precast beam installation: details in Figure 11-5;
4. Truss installation: details in Figure 11-5;
5. Roofing beam installation: details in Figure 11-6;

Each data structure can be seen as the formal representation of a generic description coming from a construction manager as, for example, in the case of the column installation:

'*At first, the Precast Column is picked up from the truck by using the hoisting machine. They are placed side-by-side. The Hoisting Machine should be located in the middle of truck and the column installation point. The labor crew is located around the same column installation point. A lifting area is required into direct contact with the labor crew and the hoisting machine. This area is used by a skilled worker who drives the installation*'.

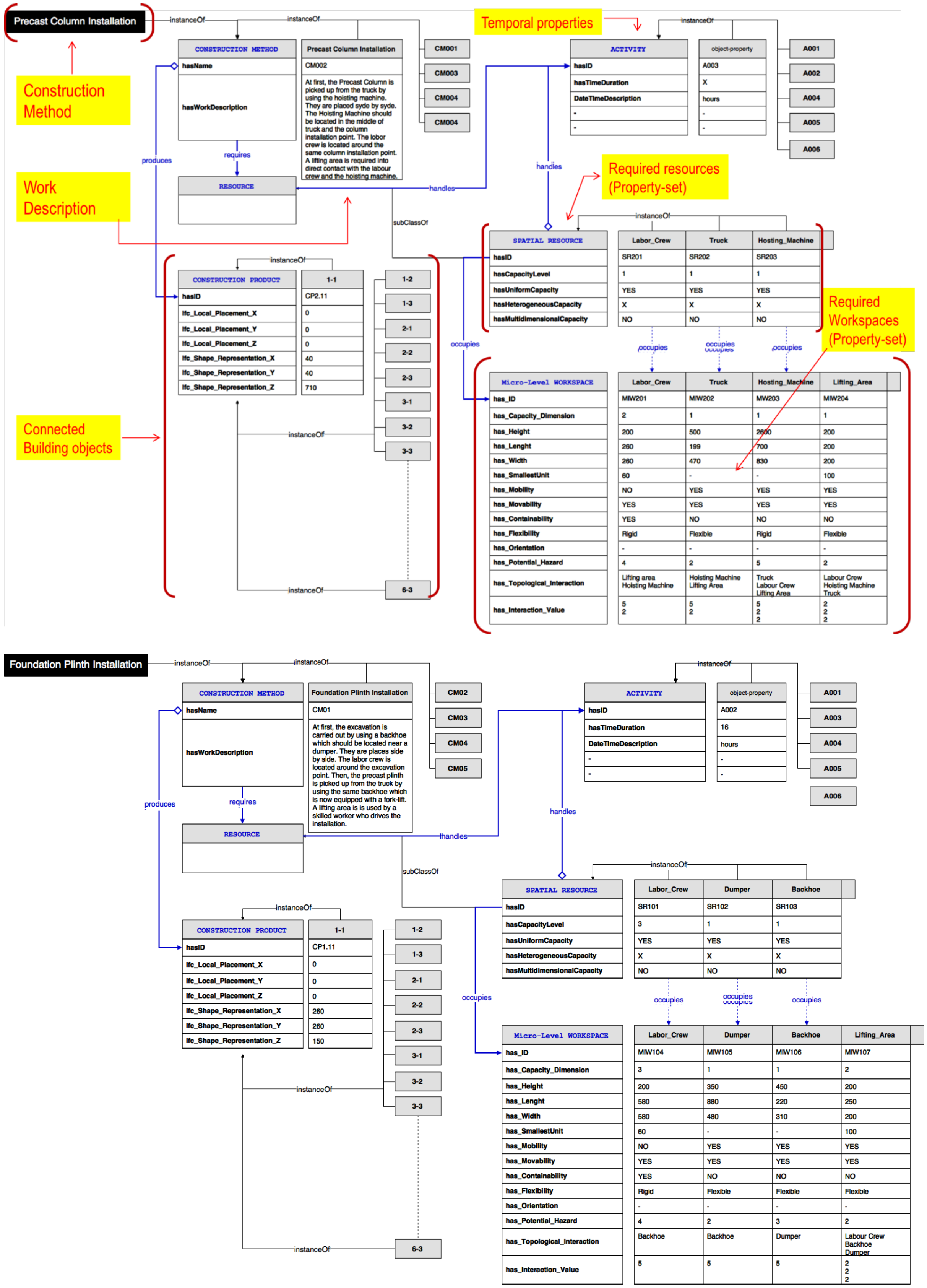


Figure 11-4 Construction Method Specifications within the ontology. Column (bottom) Plinth (top) Installation

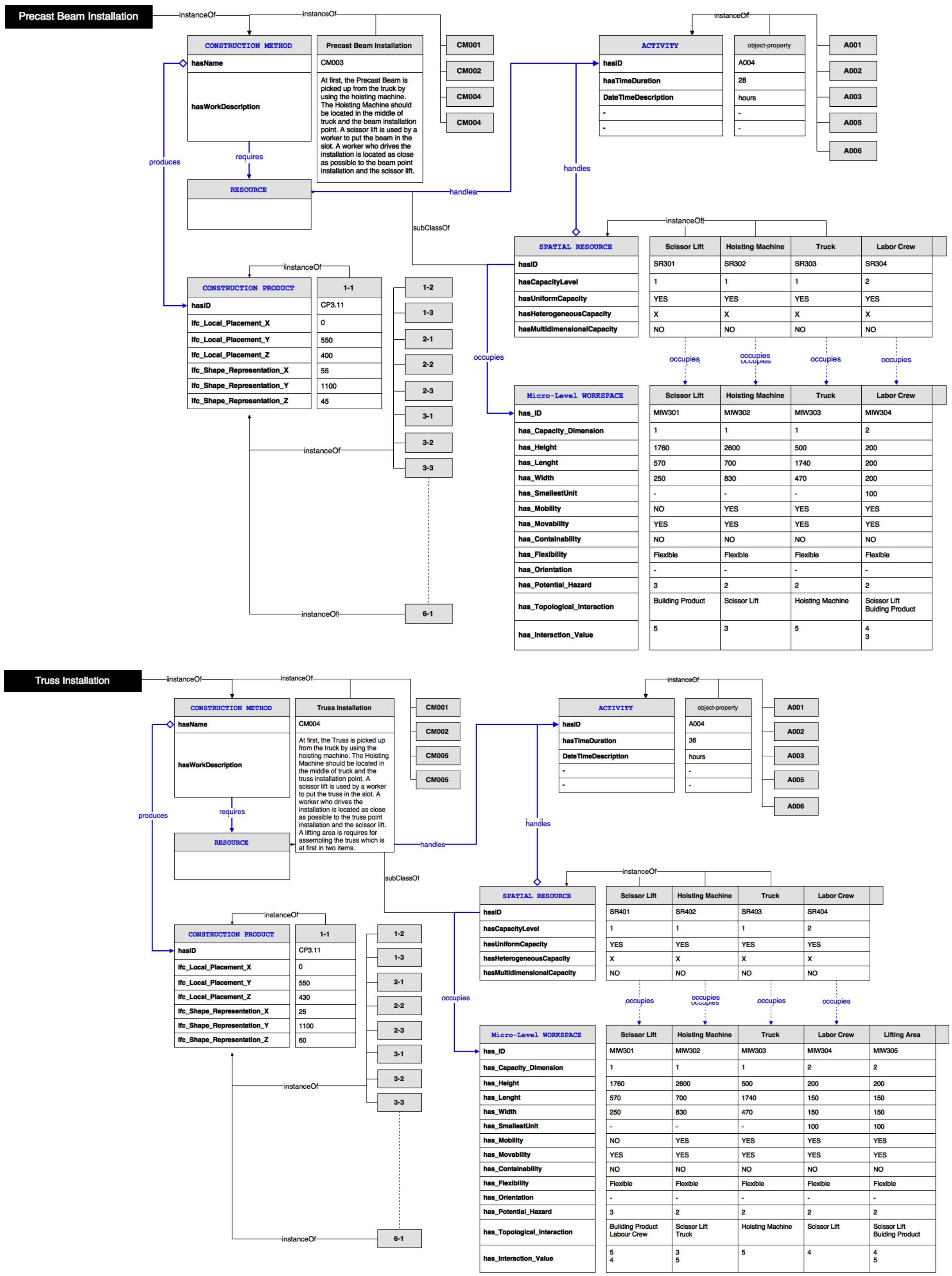


Figure 11-5 Construction Method Specifications. Precast beam (top) and Truss Installation (bottom)

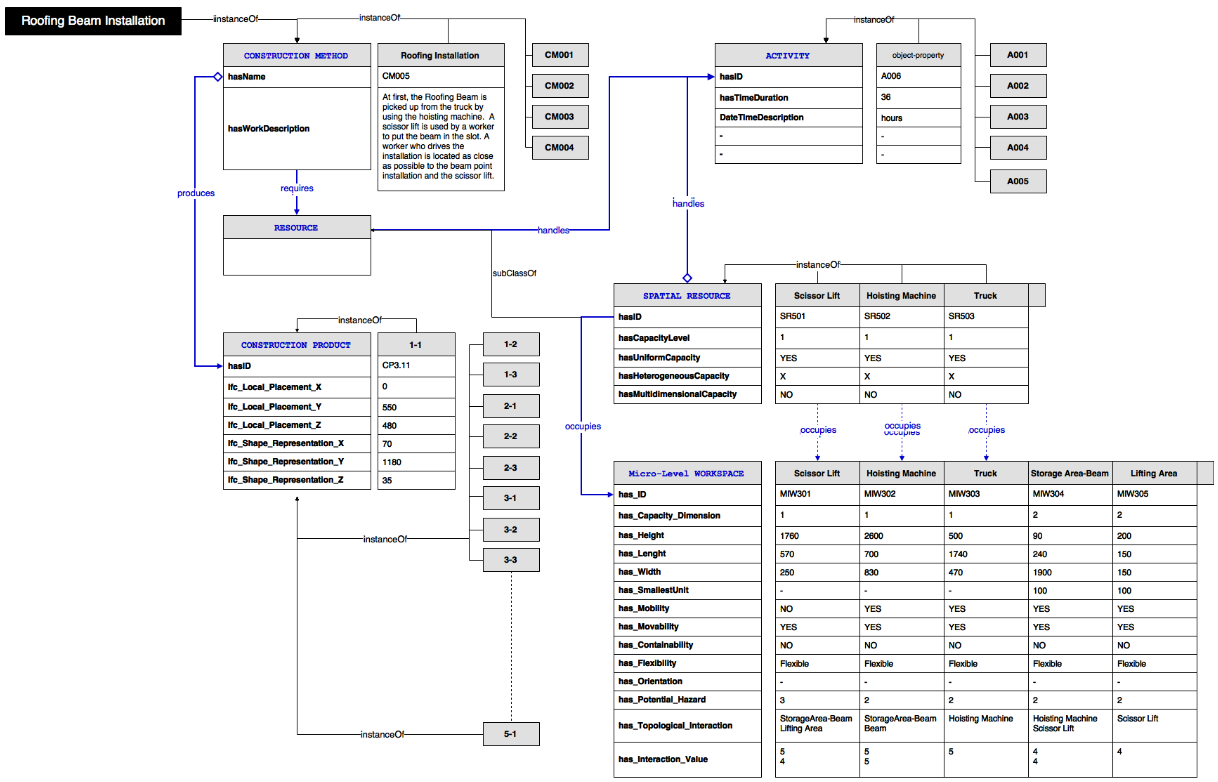


Figure 11-6 Construction Method Specifications. Case of roofing beam installation

11.2 Module 2: Bounding-box model generation

According to the *Construction Product Ontology*, the introduction of the IFC-data into the system architecture aims to reduce the building objects with their 3D shapes. The conversion from the IFC to OWL has been possible. A graphical algorithm has been developed by means of a *Dynamo's script* (Figure 11-7), an algorithm editing environment for computational design. The script is a series of written commands for generating geometries. It reads information from the Knowledge-Base and automatically sculpts the bounding boxes that surround the shapes of the building objects. By means of this integrated script, a new 'simplified-BIM' has been generated as depicted in Figure 11-8.

The same script was used to generate the workspaces geometries and to assign properties to such geometries in terms of *scheduling data* (temporal properties) -provided by the rule-engine- and *allocation data* (spatial properties) –provided by the Grasshopper's script presented in *Chapter 10*, the results of which are presented in the *Module-4*.

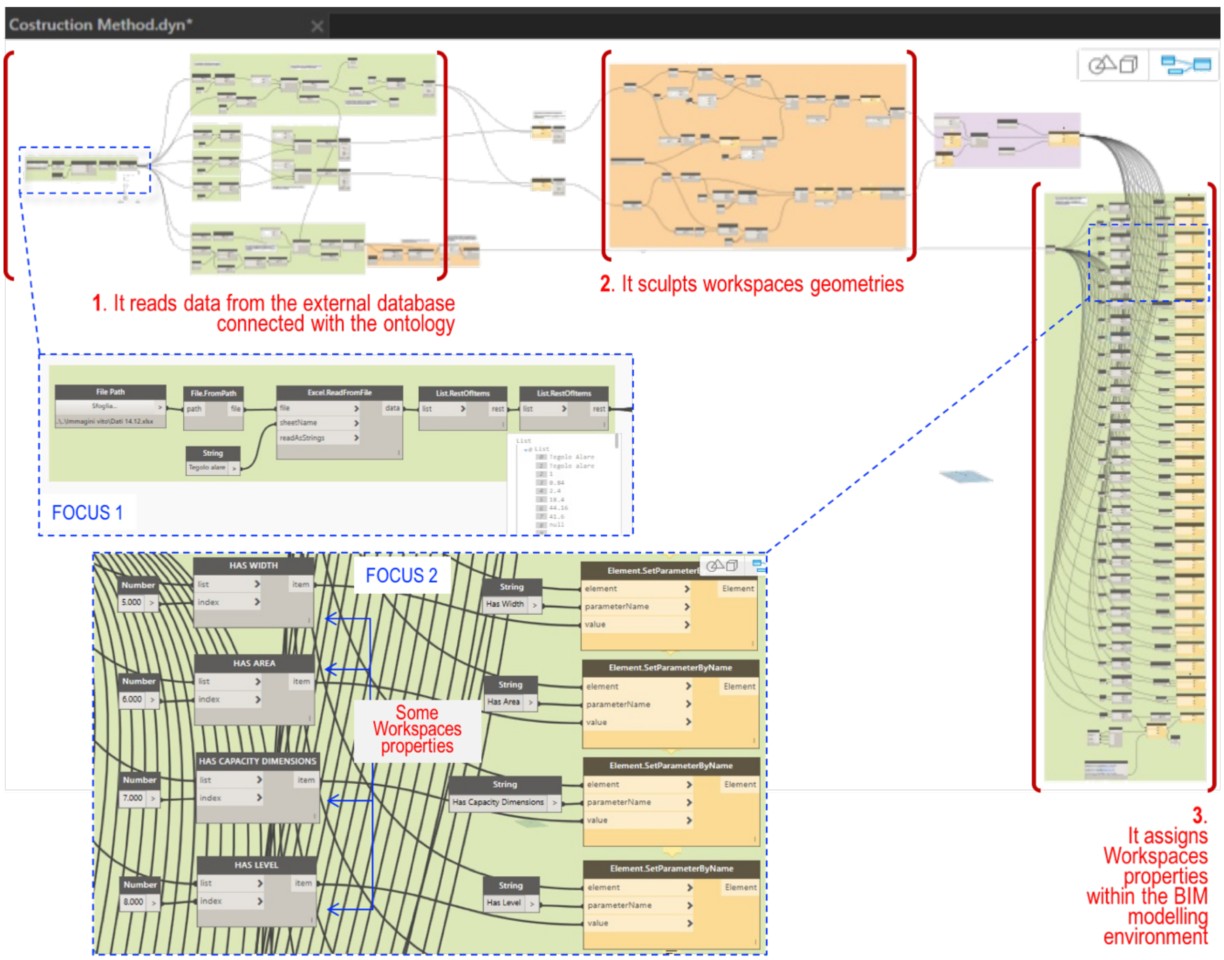
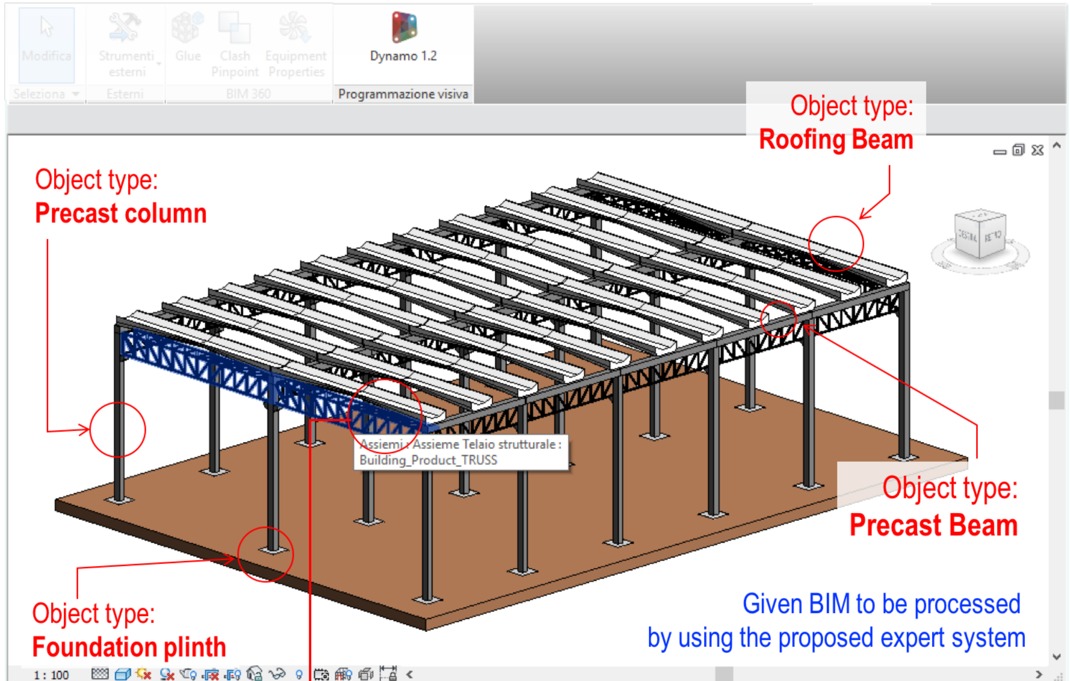


Figure 11-7 Built-in algorithm, developed by means of a *Dynamo's script* (A BIM-compliant programming environment), able to automatically sculpts geometries of building objects and workspaces in their configuration pattern by using the spatial coordinates provided by the space syntax analysis and temporal data provided by the ontology-based rule-engine.

Visualization of the given BIM to be processed



Bounding Box Model (OnSITEsimu)

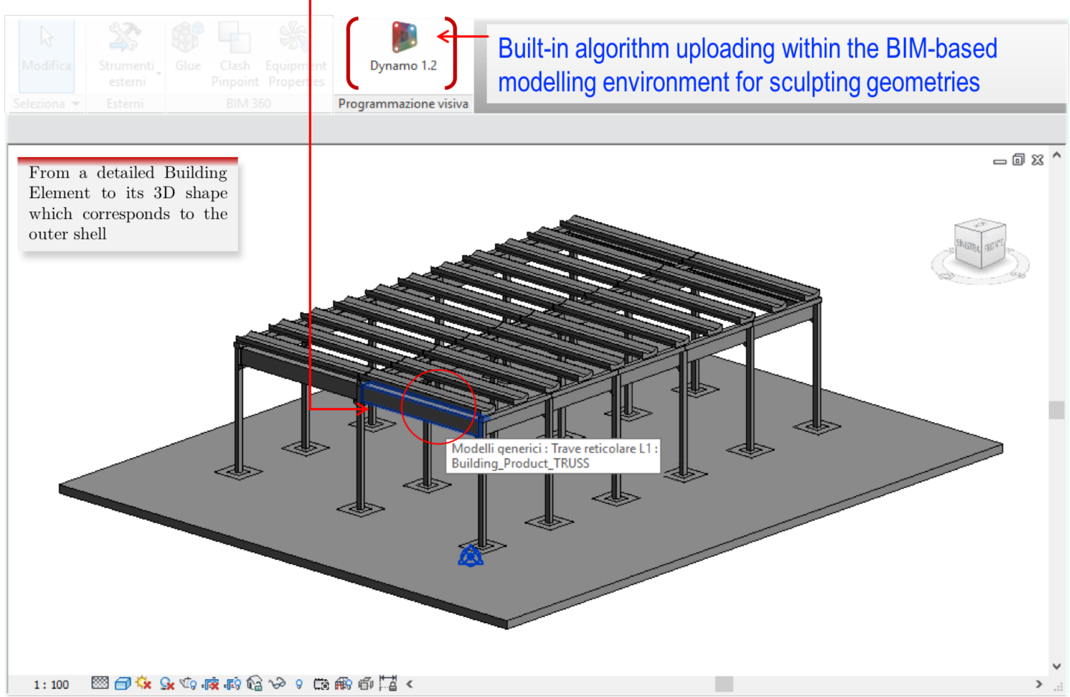
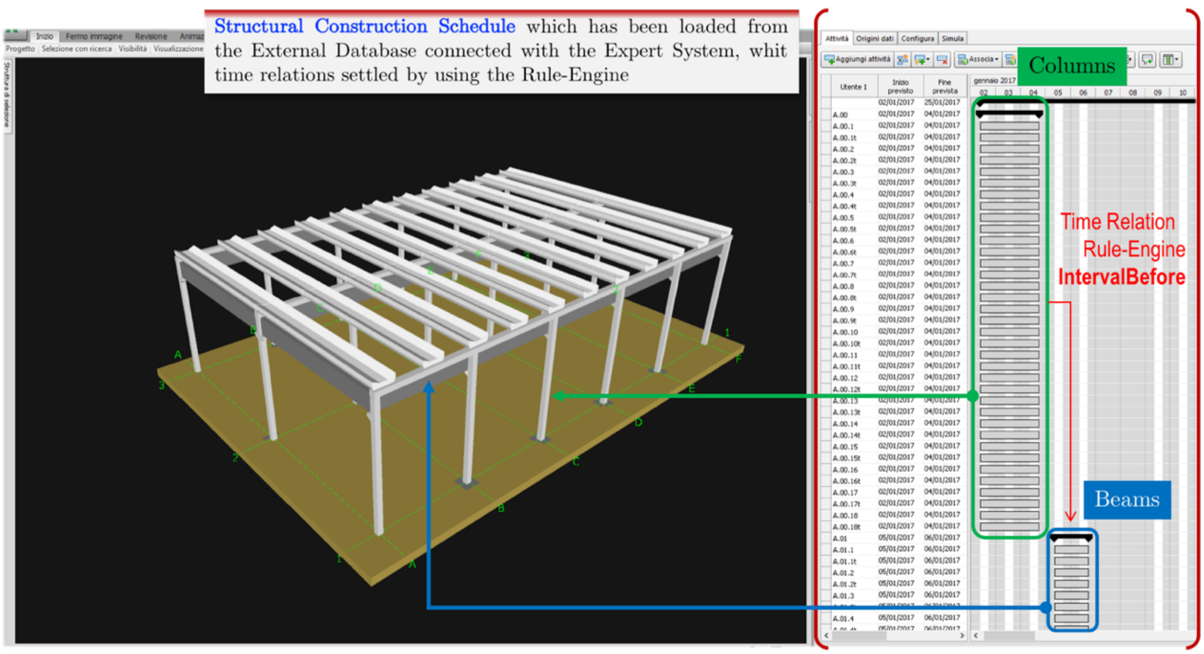


Figure 11-8 Snippet from the BIM application (*Autodesk Revit*) dynamically connected with *Dynamo* where the built-in algorithm (Figure 11-7) able to automatically sculpt geometries is stored. On the top side the visualization of given BIM before it was managed by the system. On the bottom side the 'Bounding Box model' automatically generated where it is possible to note how the representation of the beam changes

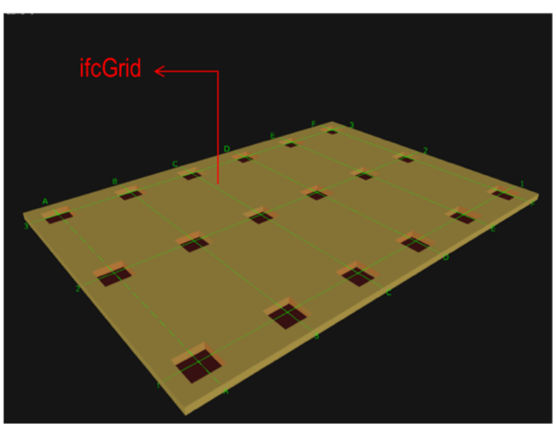
11.3 Module 3: Validation of the Structural Construction Sequence

Once the IFC data structure has been imported in the knowledge base the system operates, by means of the Rule-Engine, establishing new time relations between building objects defining the structural sequence. The Scenario 3.3 –SWRL rule- has been activated in. It updates data in the ontology with physical precedence among building products and generates new time relationships among entities. The new knowledge has been extracted and visualized by using a 4D Simulation. The main shots taken from the simulation are presented below. The structural sequence is correct and the operation of reasoning mechanism has been considered validated.

Output of the *Rule Engine* - Level 1: Simulation of the **Structural Sequence**



Construction Day: 1st



Construction Day: 4th

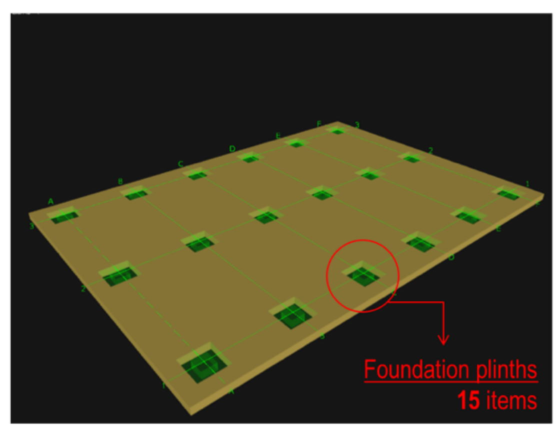


Figure 11-9 4D Simulation of the structural sequence as suggested by the KB after operating the rule-engine (1)

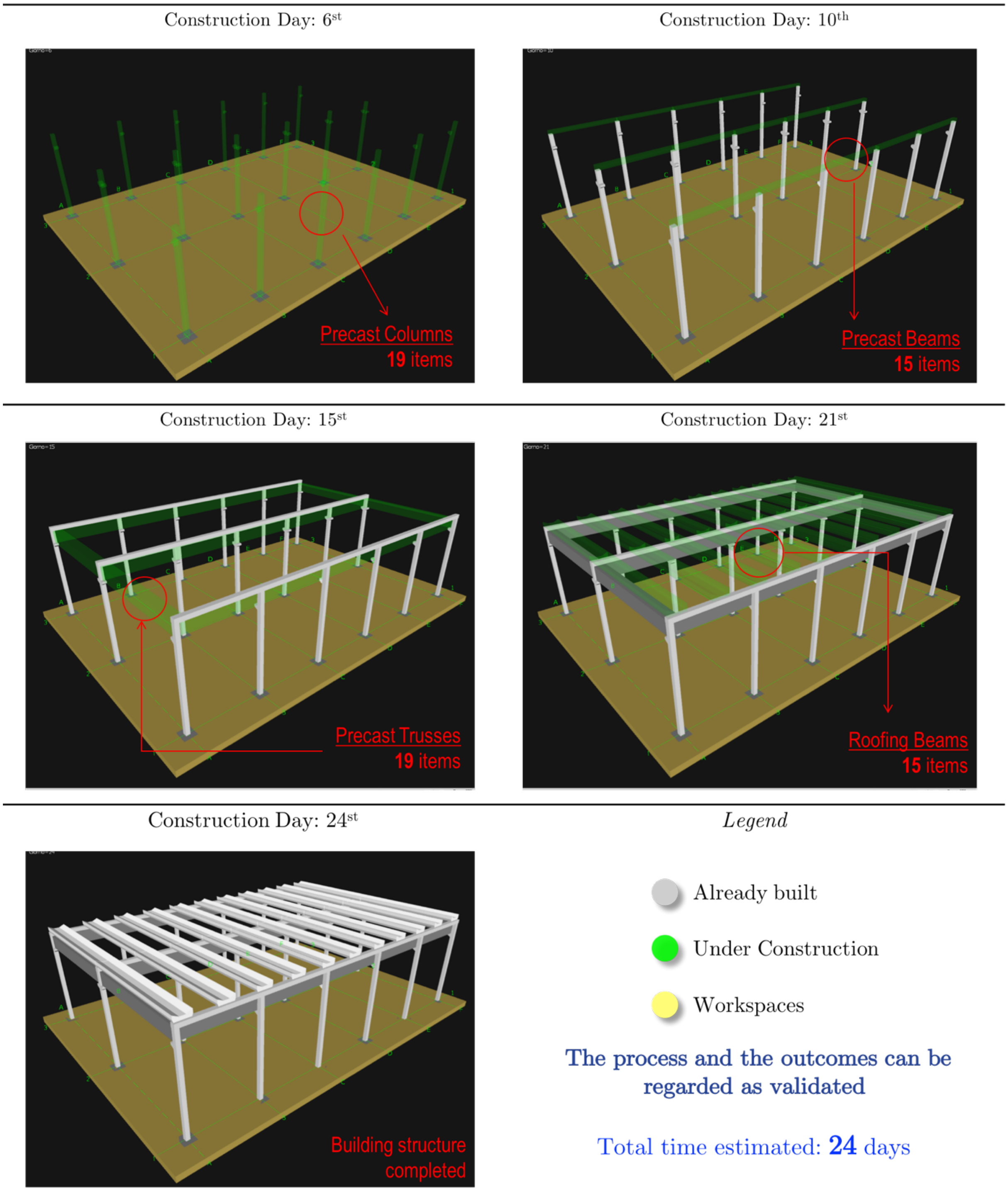


Figure 11-10 4D Simulation of the structural sequence as suggested by the KB after operating the rule-engine (2)

11.4 Module 4: Workspace planning process

The following figures depicts the results of the workspaces planning process (*Chapter 10*).

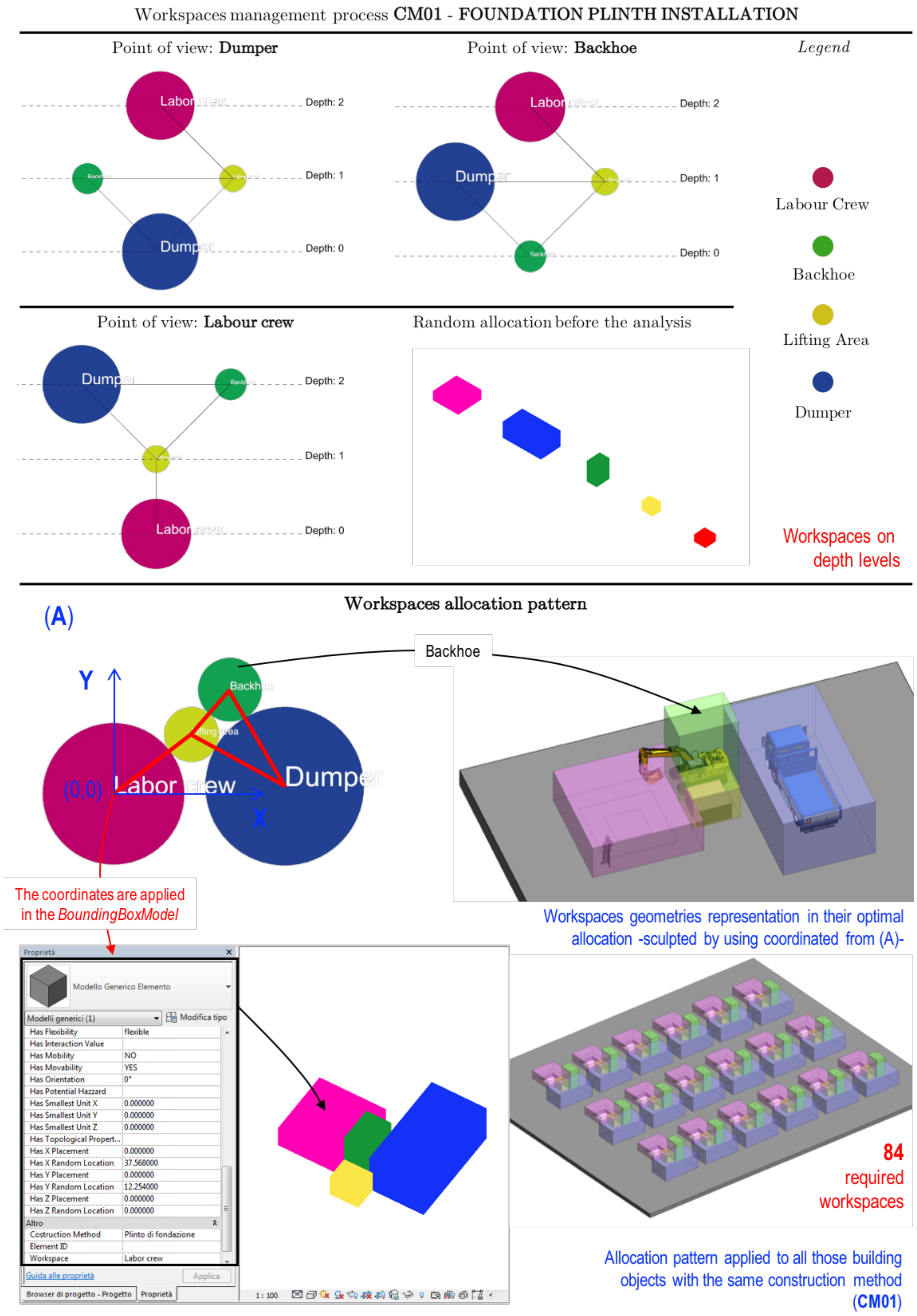


Figure 11-11 Results of the Space syntax analysis and workspaces allocation pattern for the plinths installation

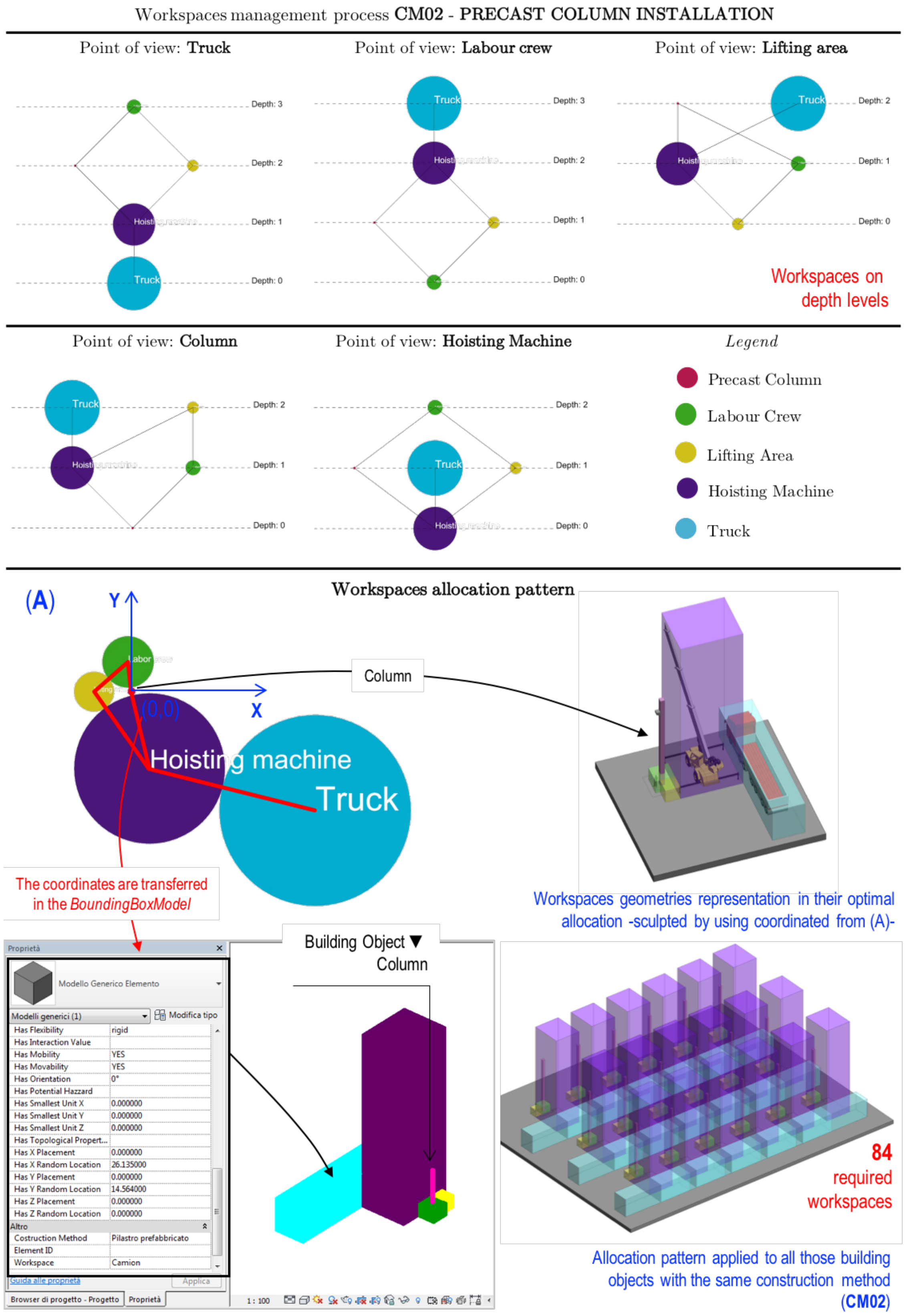


Figure 11-12 Results of the Space syntax analysis and workspaces allocation pattern for the columns installation

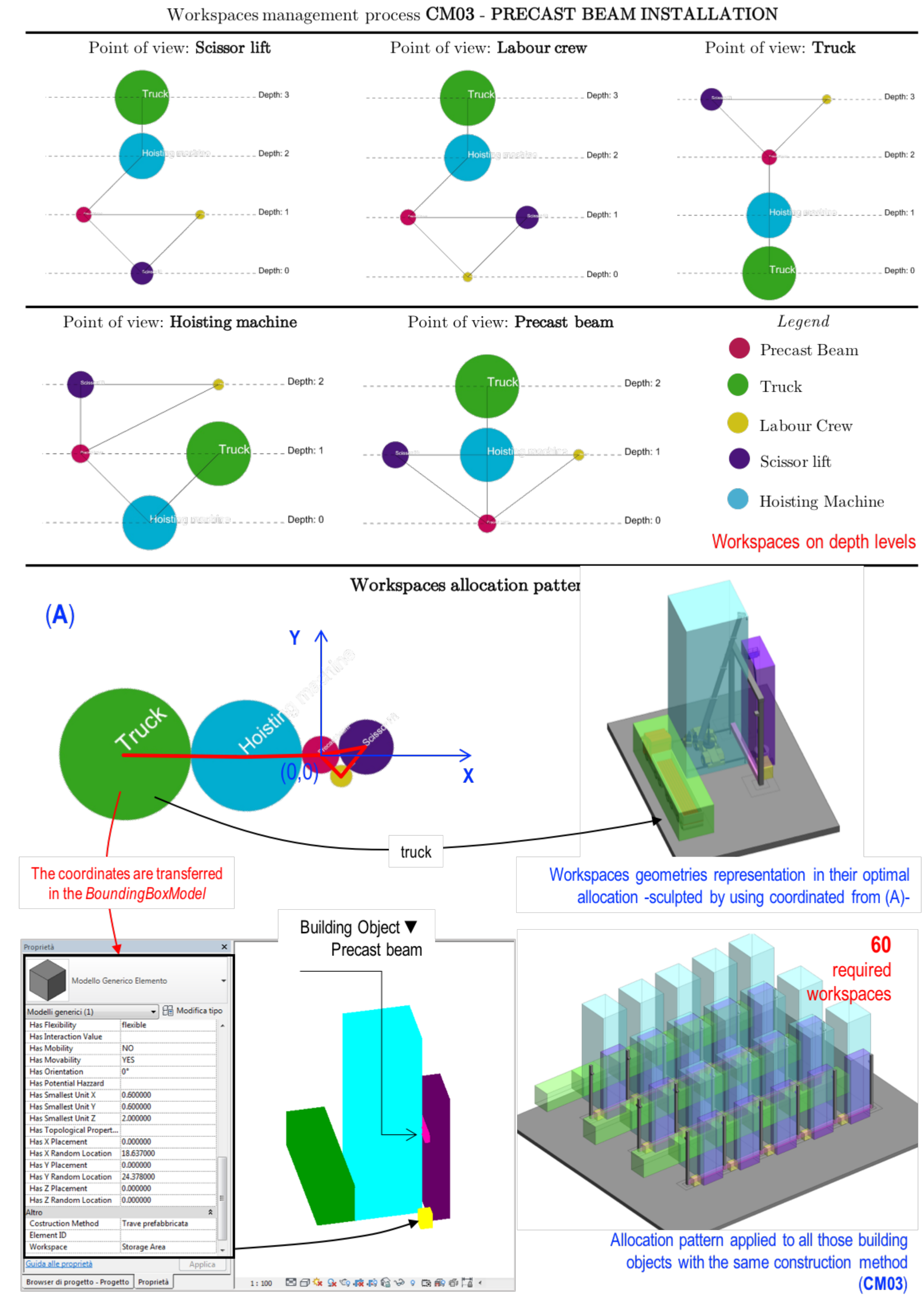


Figure 11-13 Results of the Space syntax analysis and workspaces allocation pattern for the beams installation

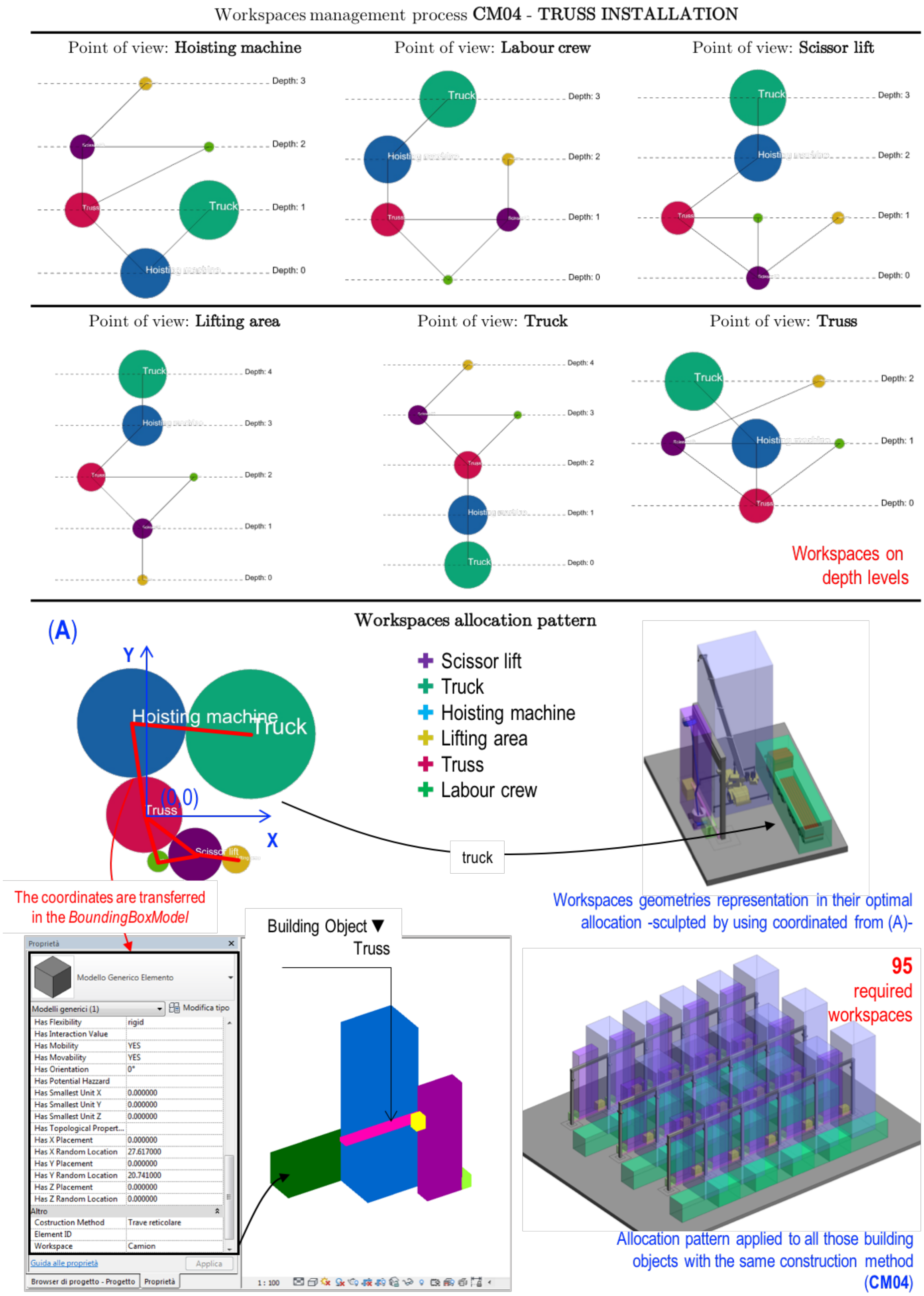


Figure 11-14 Results of the Space syntax analysis and workspaces allocation pattern for the trusses installation

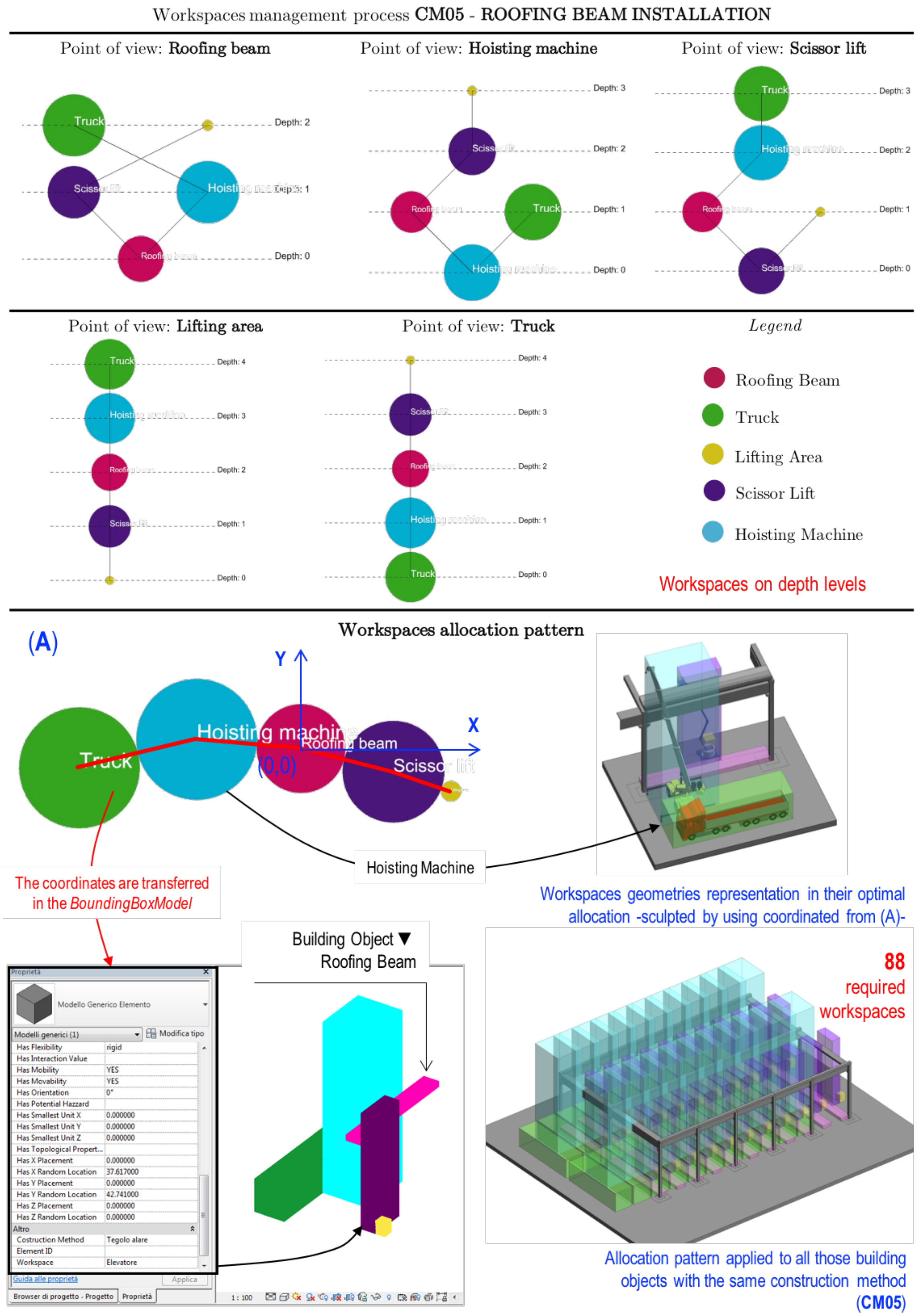


Figure 11-15 Results of the Spaces analysis and workspaces allocation pattern for the roofing beams installation

11.5 Module 5: Validation of the 'full bounding box model'

Once a workspaces configuration patterns are deducted, the spatial allocation coordinates -on the X-axis and Y-axis- of each bubble diagram has been used to allocate the workspaces at the same height (Z-axis) of their connected building object. The fully loaded bounding box model includes 611 spaces as the 4D model later presents. It is ready to be used as basis for the workspaces conflicts checking process as shown in the next paragraphs.

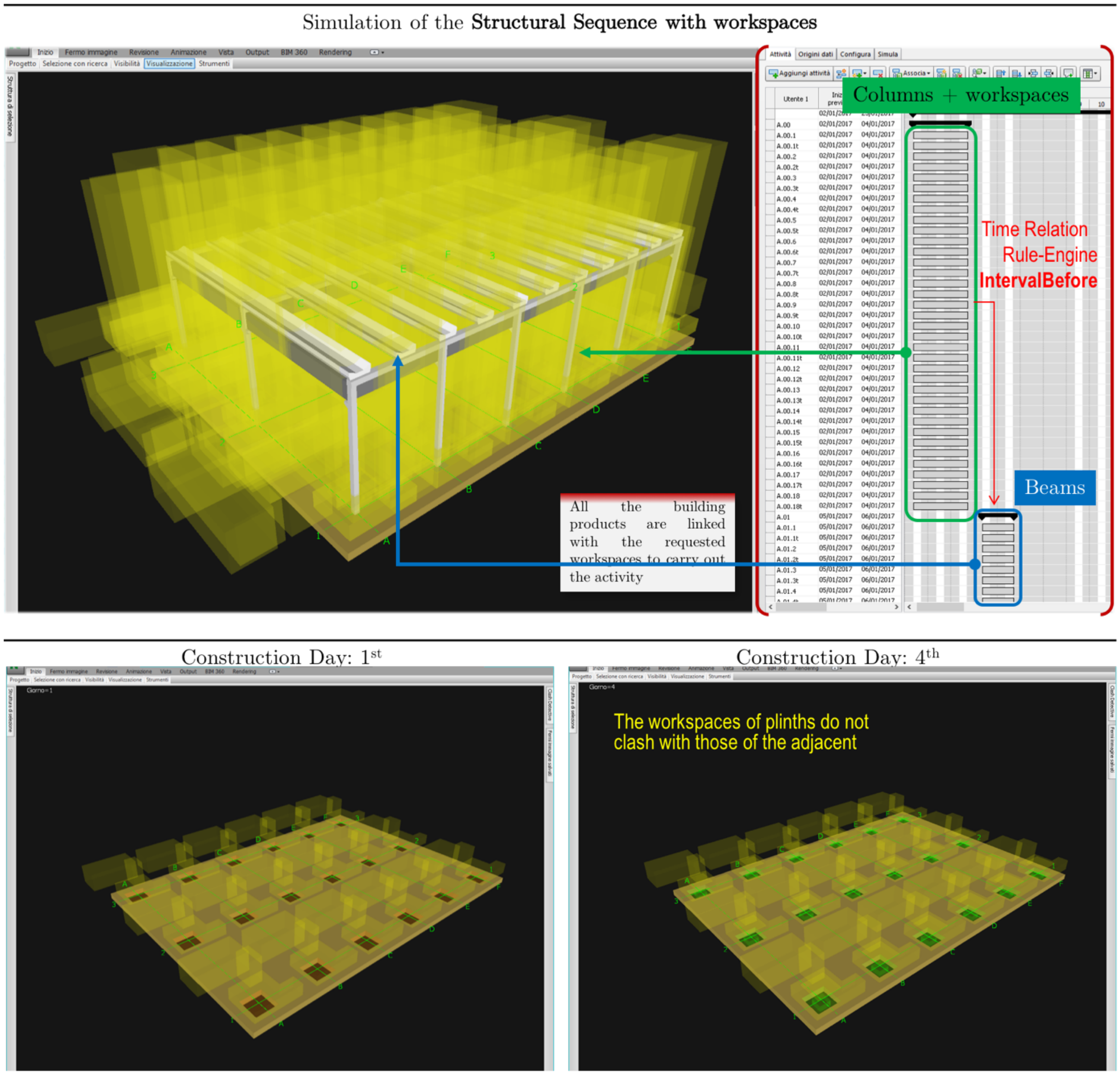


Figure 11-16 Full 'Bounding Box Model' which includes 96 building objects and 611 workspaces in their optimal layout configuration. The sub-pictures depict the structural sequence by using a 4D Simulation in a BIM environment (*Navisworks*) as suggested by the KB after operating the rule-engine. Each building objects have been linked with the required workspaces and their start and end date as well as the time period are equal (a)

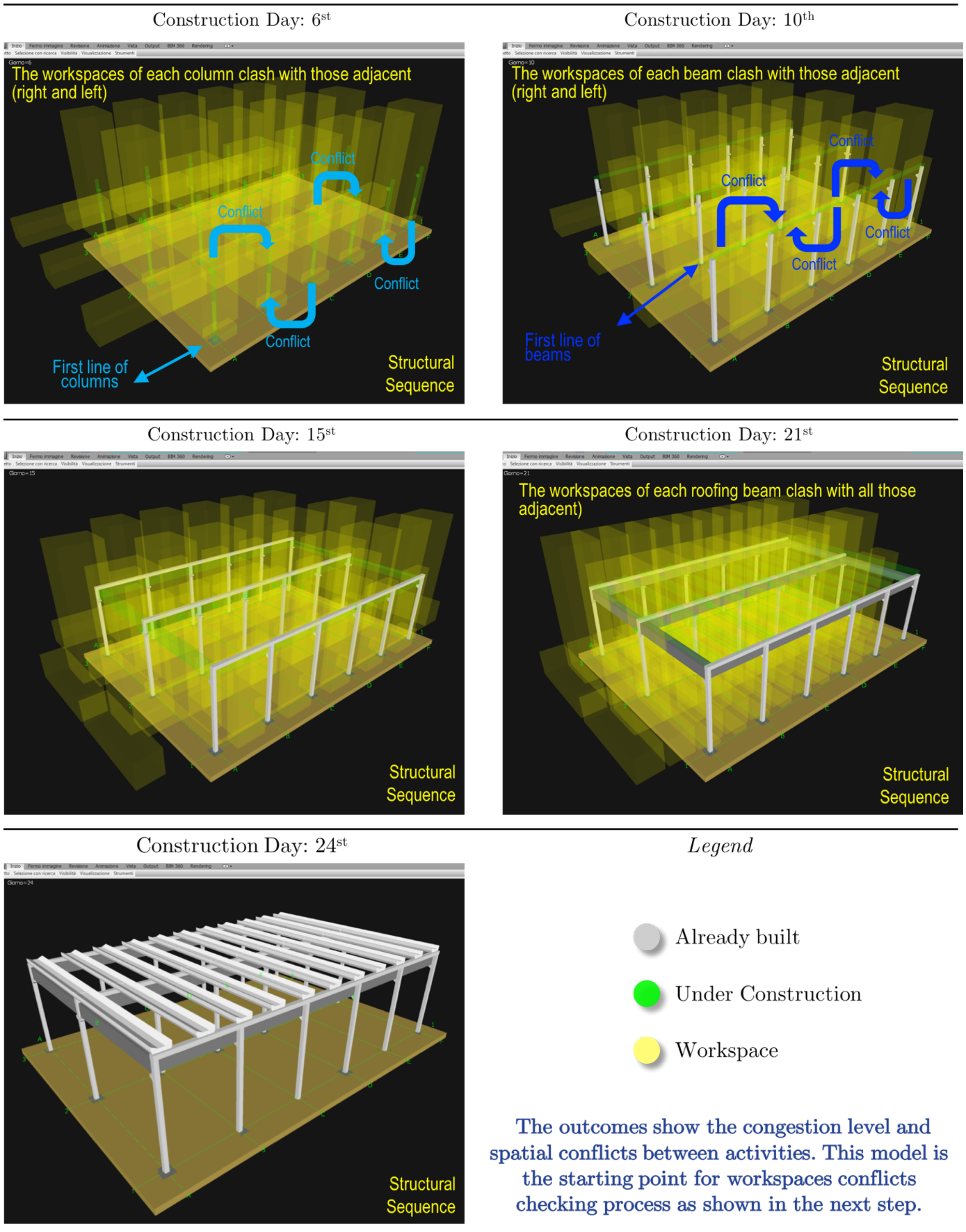


Figure 11-17 Full Bounding Box Model: 4D BIM-based Simulation of the structural sequence. (b)

11.6 Module 6: Results of the workspaces conflicts checking process

Following the loading of workspaces within the 4D BIM environment with the structural construction sequence (see precedent module) the spatial-temporal conflict detection is automatically executed loading the schedule in *Navisworks*. The process has detected 118 conflicts. The Report Tab, which includes `Item ID`, `Clash Point`, `Start Date`, `End Date`, `Distance`, `Grid Location` for each clash is fully presented in the (*Annex 1*).

The figure below shows some results for six selected conflicts. Tab. 6 summarizes the report and those data are visualized by using a bubble diagram in Figure 11-20 where each class of conflicts –filtered in terms of conflict dimension- are visualized by using a different color. This is carried out to visualize dimensions of conflicts on the building structural grid having a clearer overview of the congested areas.

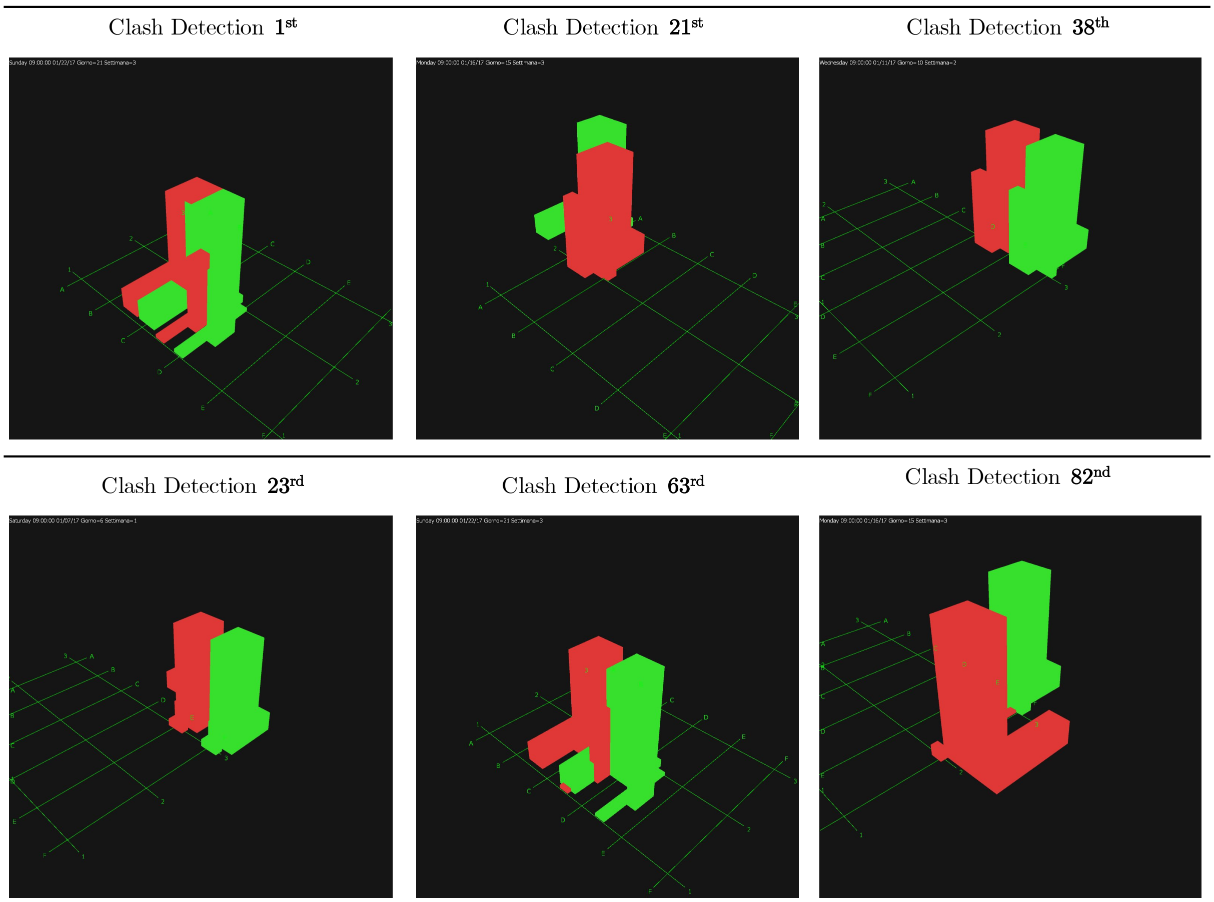


Figure 11-18 Detection and visualization of the workspaces conflicts. Each sub-picture identifies a conflict. 118 conflicts have been detected as fully depicted in the 'Clash report' (Annex 1)

Conflict	Overlapping Act.		Size	Location	
	ID 1	ID 2	[m]	X	Y
1	CM06.1	CM06.2	11,080	7,05	12,20
2	CM06.2	CM06.3	11,080	12,37	12,20
3	CM06.4	CM06.3	11,080	12,83	20,50
4	CM06.4	CM06.5	11,080	23,01	12,20
5	CM06.5	CM06.6	11,080	27,55	12,20
6	CM06.6	CM06.7	11,080	32,87	12,20
7	CM06.7	CM06.8	11,080	38,19	12,20
8	CM06.8	CM06.9	11,080	43,51	12,20
9	CM06.10	CM06.9	11,080	44,75	20,50
10	CM06.10	CM06.11	11,080	54,15	12,20
11	CM06.13	CM06.12	11,080	2,19	39,12
12	CM6.14	CM06.13	11,080	7,51	39,12
13	CM06.15	CM06.14	11,080	12,83	39,12
14	CM06.16	CM06.15	11,080	18,15	39,12
15	CM06.17	CM06.16	11,080	23,47	39,12
16	CM06.17	CM06.18	11,080	32,87	30,82
17	CM06.18	CM06.19	11,080	38,19	30,82
18	CM06.19	CM06.20	11,080	43,51	30,82
19	CM06.20	CM06.21	11,080	48,83	30,82
20	CM06.22	CM06.21	11,080	50,07	39,12
21	CM05.6	CM05.18	10,428	8,17	28,76
22	CM05.17	CM05.1	10,428	8,17	5,66
23	CM02.2	CM02.1	8,900	9,87	11,71
24	CM02.3	CM02.2	8,900	20,87	11,71
25	CM02.3	CM02.4	8,900	24,87	11,71
26	CM02.5	CM02.4	8,900	33,97	16,41
27	CM02.5	CM02.6	8,900	46,87	11,71
28	CM02.8	CM02.7	8,900	9,87	30,21
29	CM02.9	CM02.8	8,900	20,87	30,21
30	CM02.9	CM02.10	8,900	24,87	30,21
31	CM02.11	CM02.10	8,900	33,97	34,91
32	CM02.12	CM02.11	8,900	44,97	34,91
33	CM02.14	CM02.13	8,900	9,87	48,71
34	CM02.14	CM02.15	8,900	13,87	48,71
35	CM02.15	CM02.16	8,900	24,87	48,71
36	CM02.17	CM02.16	8,900	33,97	53,41
37	CM02.17	CM02.18	8,900	46,87	48,71
38	CM04.2	CM04.1t	7,900	15,17	2,36
39	CM04.2	CM04.3t	7,900	26,17	2,36
40	CM04.3	CM04.4	7,900	37,17	2,36
41	CM04.5	CM04.4	7,900	48,17	2,36
42	CM04.7	CM04.6	7,900	15,17	20,86
43	CM04.7	CM04.8	7,900	26,17	20,86
44	CM04.8	CM04.9	7,900	37,17	20,86
45	CM04.10	CM04.9	7,900	48,17	20,86
46	CM04.12	CM04.11	7,900	15,17	39,36
47	CM04.12	CM04.13	7,900	26,17	39,36
48	CM04.13	CM04.14	7,900	37,17	39,36
49	CM04.14	CM04.15	7,900	48,17	39,36
50	CM05.2	CM05.1	7,509	15,17	2,46
51	CM05.2	CM05.3	7,509	26,17	2,46
52	CM05.3	CM05.4	7,509	37,17	2,76
53	CM05.5	CM05.4	7,509	48,17	2,76
54	CM05.7	CM05.6	7,509	15,17	20,16
55	CM05.7	CM05.8	7,509	26,17	20,16
56	CM05.8	CM05.9	7,509	37,17	20,16

Conflict	Overlapping Act.		Size	Location	
	ID 1	ID 2	[m]	X	Y
63	CM06.4	CM06.2	5,760	13,89	14,10
64	CM06.4	CM06.6	5,760	24,53	12,20
65	CM06.5	CM06.3	5,760	19,21	12,20
66	CM06.5	CM06.7	5,760	29,07	12,20
67	CM06.6	CM06.8	5,760	34,39	12,20
68	CM06.7	CM06.9	5,760	39,71	12,20
69	CM06.10	CM06.8	5,760	45,81	12,20
70	CM06.11	CM06.9	5,760	51,13	12,20
71	CM06.14	CM06.12	5,760	8,57	32,72
72	CM06.15	CM06.13	5,760	13,89	32,72
73	CM06.16	CM06.14t	5,760	19,21	30,82
74	CM06.16	CM06.18	5,760	29,07	30,82
75	CM06.17	CM06.15	5,760	24,53	30,82
76	CM06.17	CM06.19	5,760	34,39	30,82
77	CM06.18	CM06.20	5,760	39,71	30,82
78	CM06.19	CM06.21	5,760	45,03	30,82
79	CM06.22	CM06.20	5,760	51,13	30,82
80	CM05.10	CM05.19	5,528	57,64	20,42
81	CM05.5	CM05.16	5,328	57,64	2,56
82	CM05.19	CM05.15	2,700	57,64	38,66
83	CM05.11	CM05.18	2,530	5,70	38,46
84	CM05.10	CM05.16	2,500	57,64	20,16
85	CM05.17	CM05.6	2,405	5,70	20,16
86	CM06.1	CM06.12	1,280	3,87	22,00
87	CM06.1	CM06.14	1,280	3,87	21,72
88	CM06.2	CM06.13	1,280	9,19	22,00
89	CM06.3	CM06.14	1,280	14,51	22,00
90	CM06.4	CM06.15	1,280	19,83	22,00
91	CM06.4	CM06.17	1,280	19,83	21,72
92	CM06.5	CM06.16	1,280	25,15	22,00
93	CM06.5	CM06.18	1,280	25,15	21,72
94	CM06.10	CM06.21	1,280	51,75	22,00
95	CM06.15	CM06.2	1,280	9,19	22,00
96	CM06.16	CM06.3	1,280	14,51	22,00
97	CM06.17	CM06.6	1,280	30,47	20,72
98	CM06.18	CM06.7	1,280	35,79	20,72
99	CM06.19	CM06.6	1,280	30,47	22,00
100	CM06.19	CM06.8	1,280	41,11	20,72
101	CM06.20	CM06.7	1,280	35,79	22,00
102	CM06.20	CM06.9	1,280	46,43	20,72
103	CM06.21	CM06.8	1,280	41,11	22,00
104	CM06.22	CM06.9	1,280	46,43	22,00
105	CM06.22	CM06.11	1,280	57,07	20,72
106	CM05.17	CM05.18	0,600	5,70	21,66
107	CM05.19	CM05.16	0,600	57,64	20,16
108	CM06.3	CM06.6	0,440	19,21	8,50
109	CM06.4	CM06.1	0,440	8,13	8,50
110	CM06.4	CM06.7	0,440	24,53	8,50
111	CM06.5	CM06.2	0,440	13,45	8,50
112	CM06.5	CM06.8	0,440	29,85	8,50
113	CM06.6	CM06.9	0,440	35,17	8,50
114	CM06.10	CM06.7	0,440	40,05	8,50
115	CM06.11	CM06.8	0,440	45,37	8,50
116	CM06.15	CM06.12	0,440	8,13	27,12
117	CM06.16t	CM06.13	0,440	13,45	27,12
118	CM06.16t	CM06.19	0,440	29,85	27,12

Tab. 6 Report tab of the workspaces conflicts checking process. For each clash the following data are stored: Clash number, Items ID, X and Y coordinates of the clash points on the building grid, conflict size.

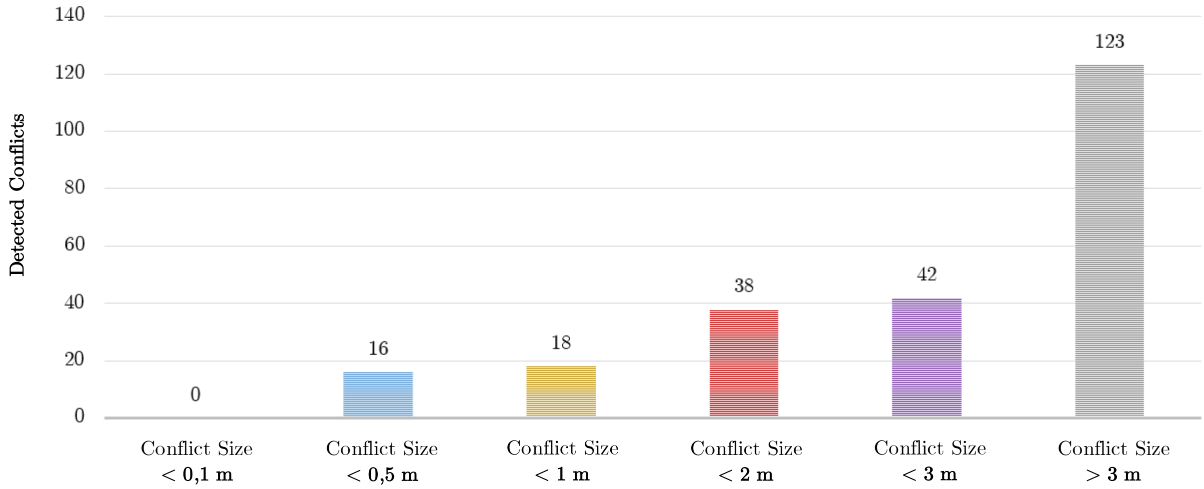


Figure 11-19 Bar Chart for displaying the amount of conflicts depending on their size

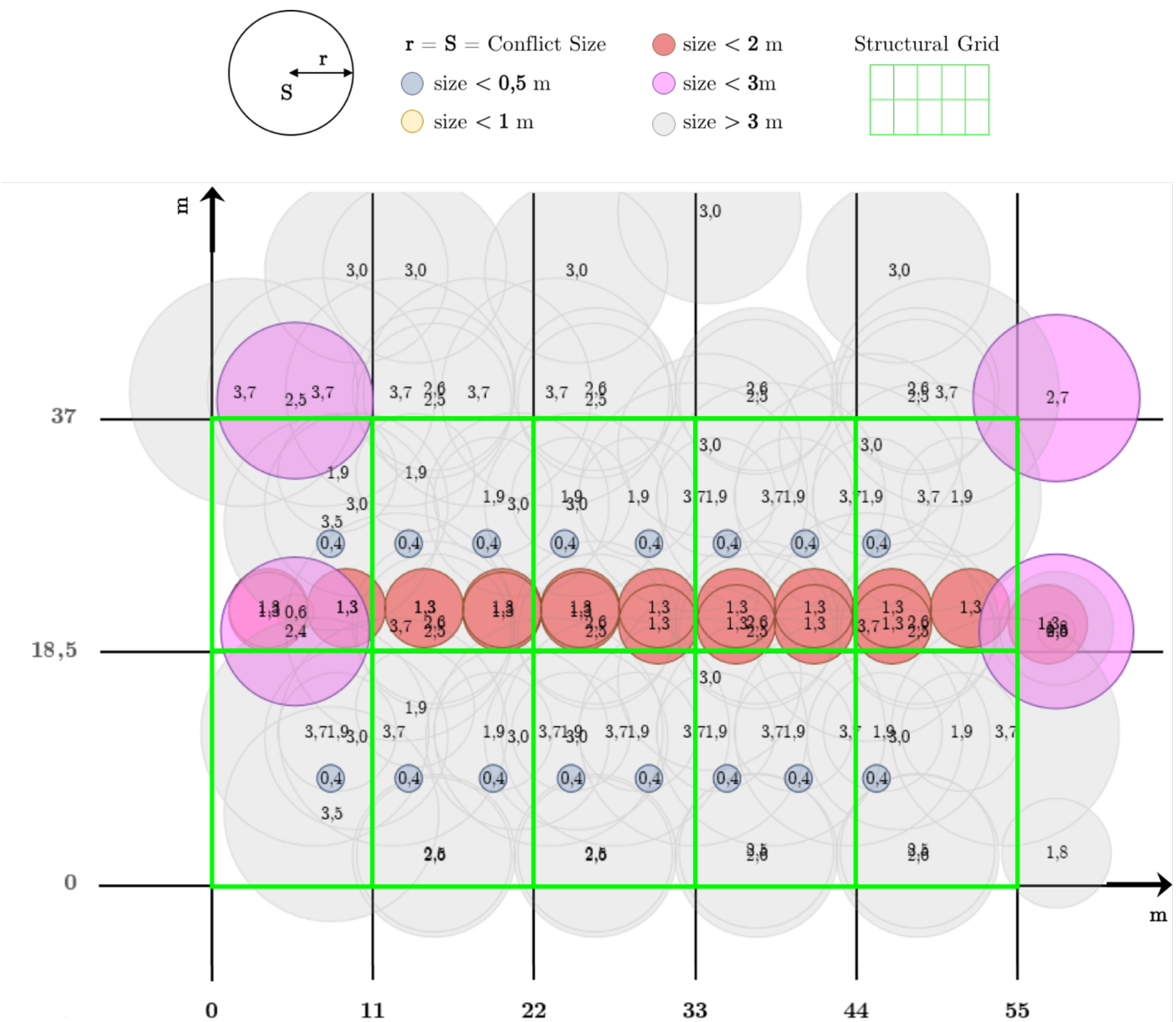


Figure 11-20 Bubble Diagram for displaying the workspaces conflicts allocation in plant with reference to the structural grid. i) the bubble color represents the conflict size according to the chart in Figure 11-19

11.7 Module 7: Validation of the shortest completion sequence

After results of the previous modules, the system proceeds by expanding the first schedule (structural sequence). Conflicts, stored in the Centralized Database (Connor, 2009), become temporal constraints in the *Construction Scheduling Ontology*. Everyone is converted in a new temporal relationship (*intervalBefore*) between all those entities linked with the detected workspaces (e.g., activities, resources, workspaces, building products, etcetera). This is, once again, carried out by using the SWRL rule-set. The results, for the rules validation, are shown in the pictures below. Attention is drawn to the fact that the construction time duration gets up from 24 to 57 days.

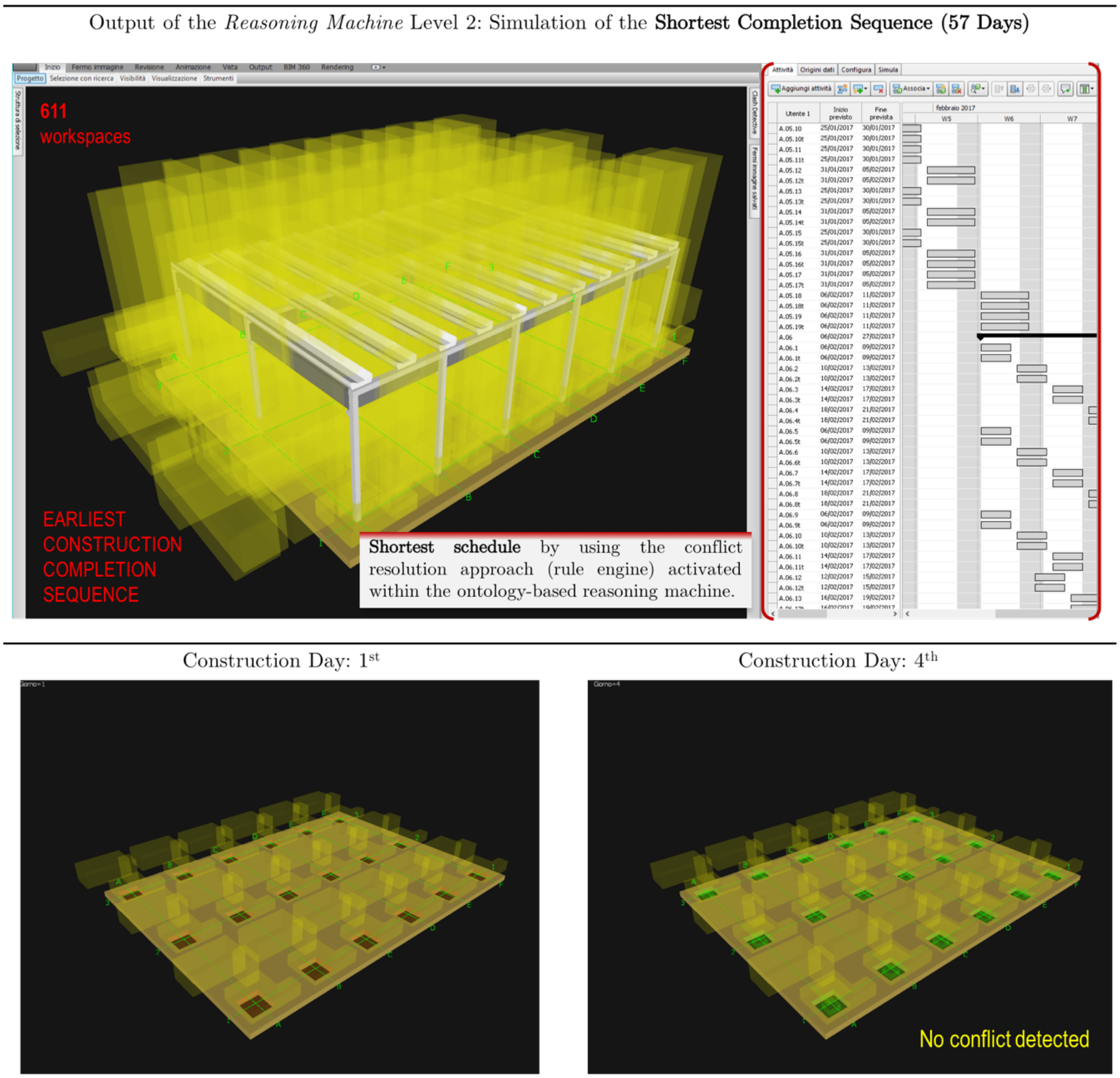


Figure 11-21 4D Simulation of the earliest construction sequence as suggested by the KB after operating the rule-engine. The workspaces conflicts have been solved establishing new time relations between overlapping entities (i)

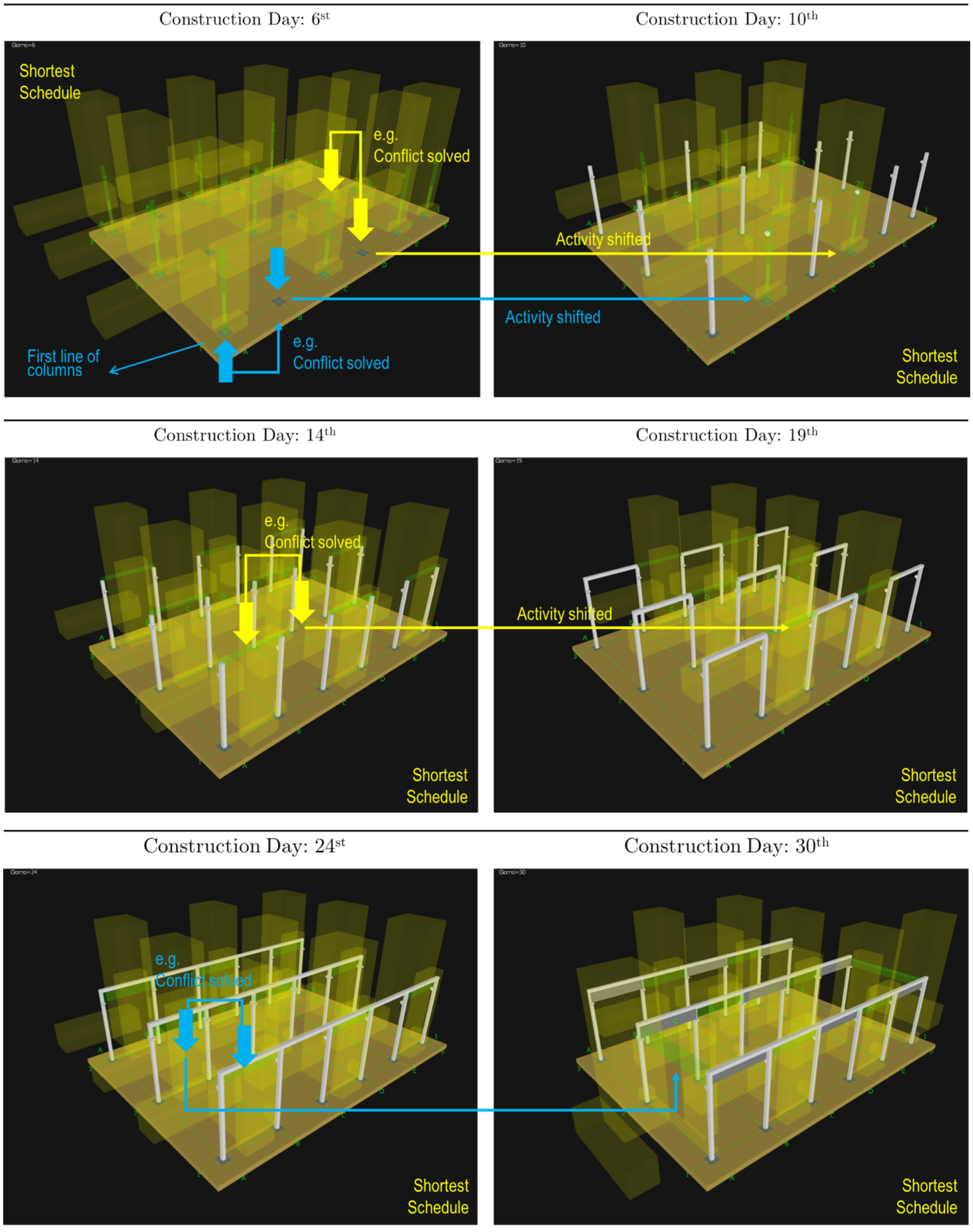


Figure 11-22 4D BIM-based simulation of the earliest construction sequence. From the 6st to the 30th day

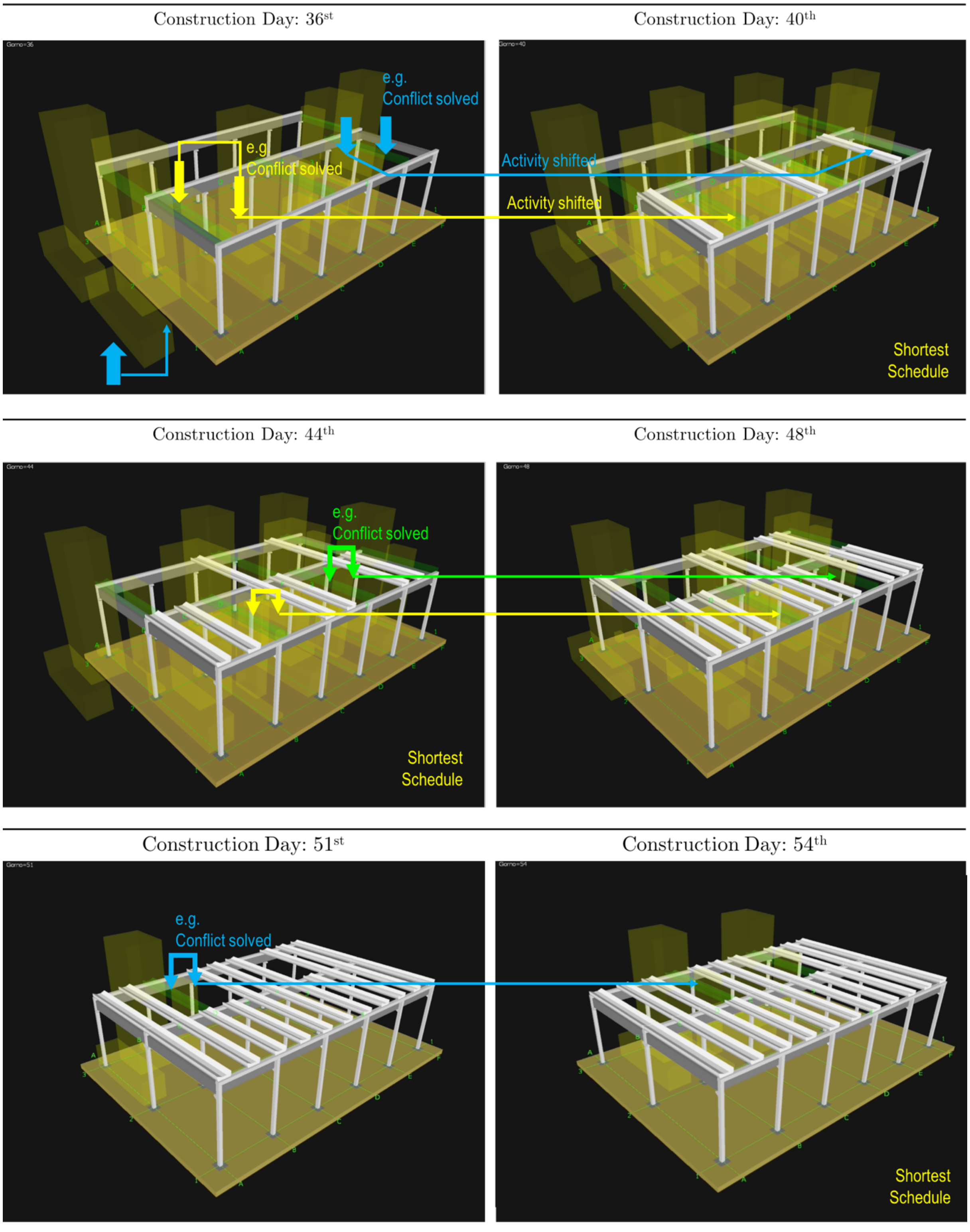


Figure 11-23 4D Simulation of the earliest construction sequence. From the 36^{st} to the 54^{th} day

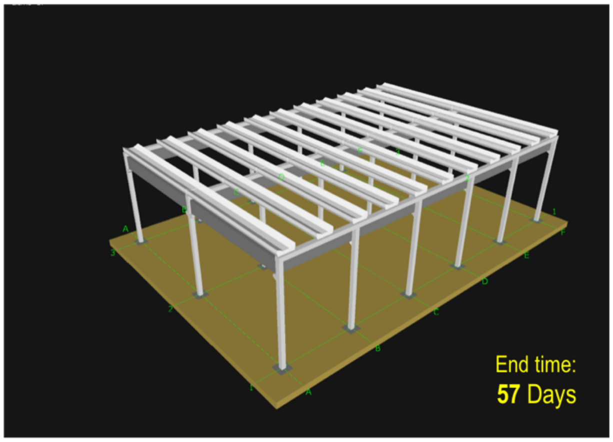
Construction Day: **57**st	*Legend*
	Already built Under Construction Workspace

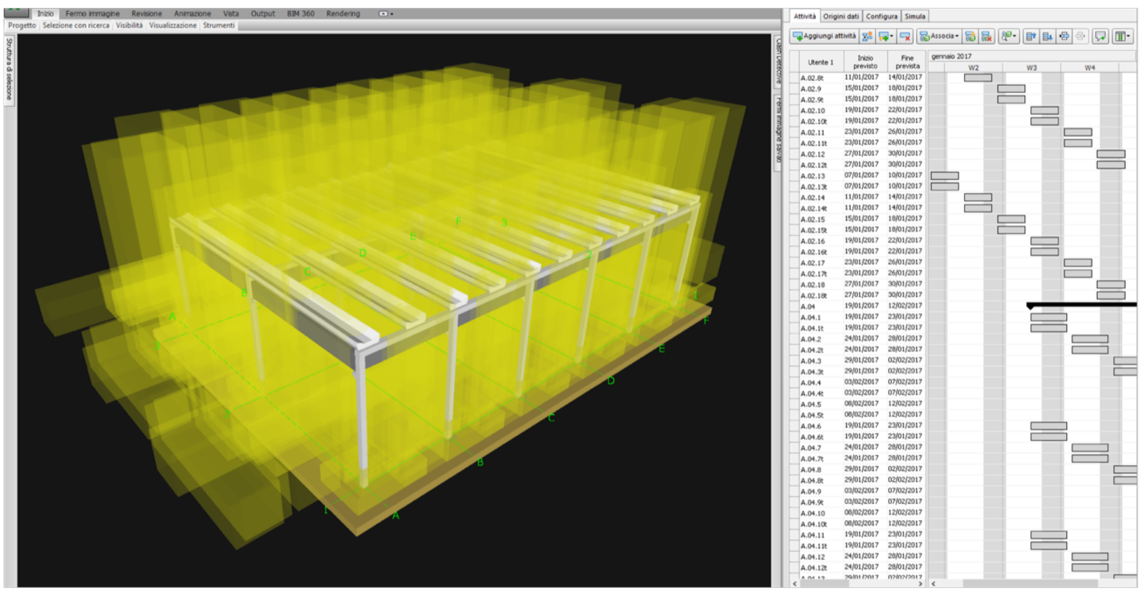
Figure 11-24 Simulation of the end date of the earliest construction completion sequence

In order to validate the process and the truth of outcomes a new spatial-temporal conflict detection has been executed by using the same BIM Simulation Environment. No conflict has been detected.

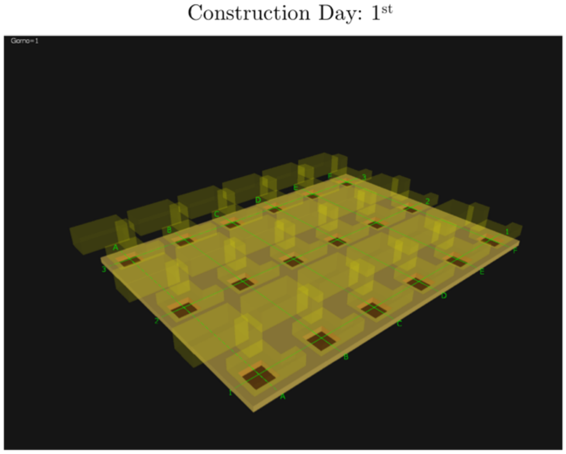
11.8 Comparison of schedules: construction sequence bottom-up, left-to-right

According to the original objectives, the system is on one hand equipped with a number of rules, but new ones can be configured and adjusted by the user. For this reason, a different strategy –rule- (see Par. 2 - *Chapter 10*) has been added in order to verify if and how the schedule changes when a different construction strategy/sequence is loaded allowing the system to generate the earliest construction schedule which fulfils the strategy itself. The output by using a construction sequence bottom-up and left-to-right is shown in the pictures below. Attention is drawn to the fact that the construction time duration gets up from 57 (earliest construction sequence suggested by the system) to 117 days.

Output of the *Reasoning Machine* Level 3: **Construction Sequence *Bottom-up left-to-right*** **(117 Days)**



Construction Day: 1st



Construction Day: 4th

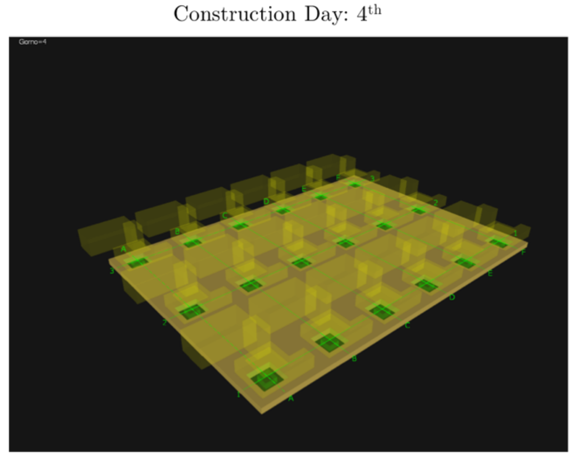


Figure 11-25 4D Simulation of the construction sequence bottom-up and left-to-tight. From the 1st to the 4th day.

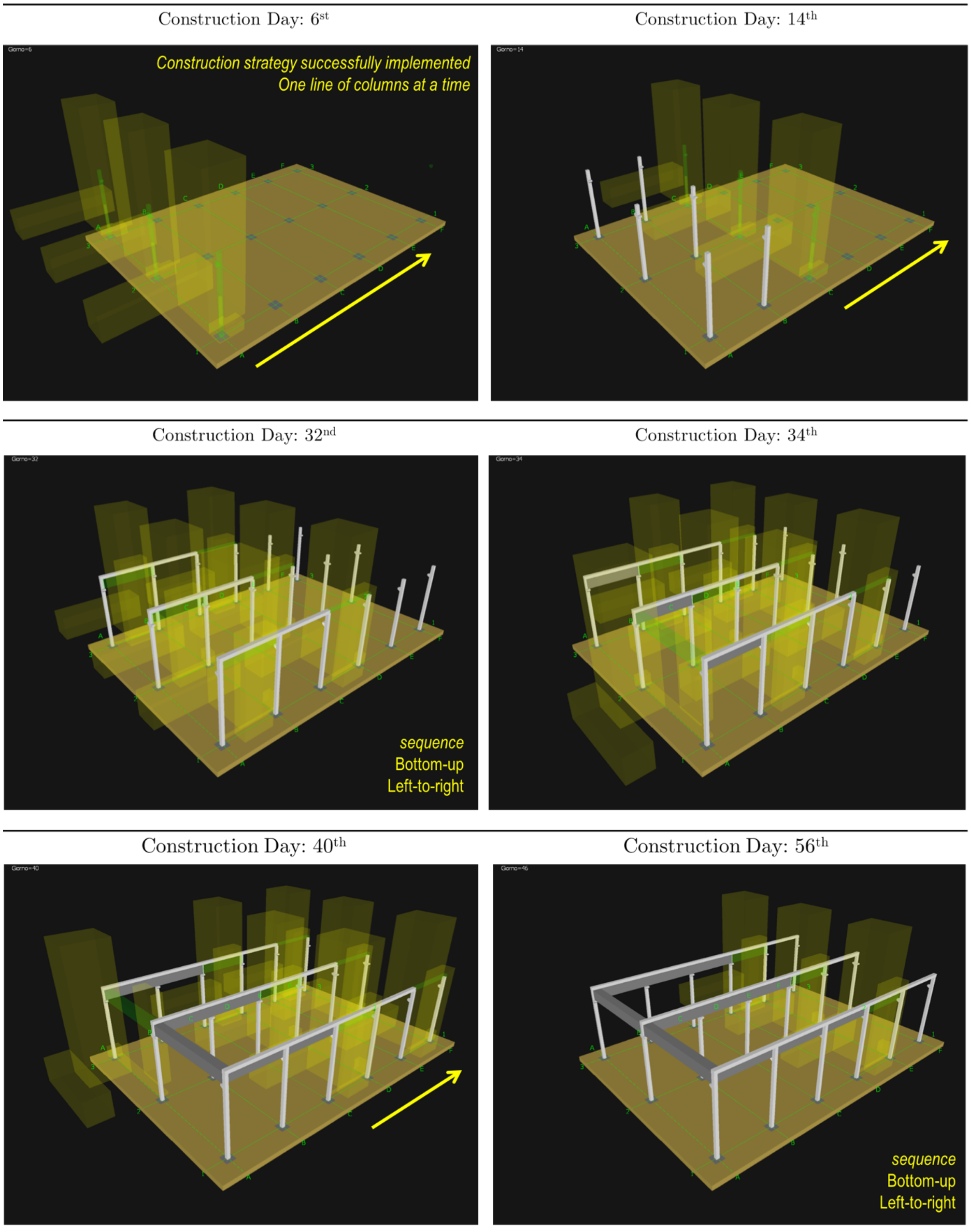


Figure 11-26 4D Simulation: sequence '*bottom-up* and *left-to-tight*'. From the 6st to the 56th day.

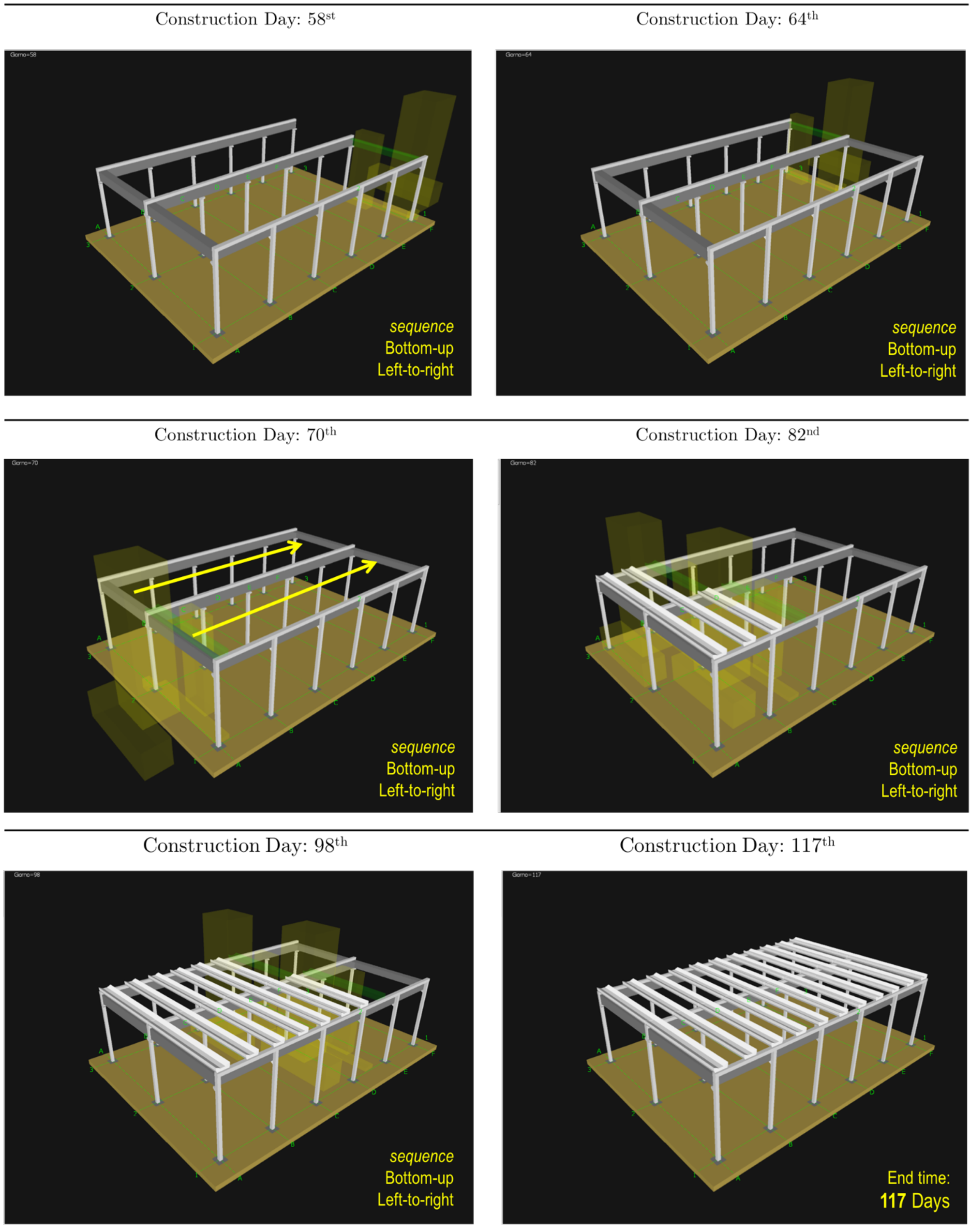


Figure 11-27 4D Simulation: sequence '*bottom-up* and *left-to-tight'*. From the 58st to the 117th day.

Chapter 12 Concluding remarks

12.1 Research contributions

With the increasing complexity of large construction projects, the need of automation in construction schedules design and generation became increasing in construction management in order to improve site productivity.
The study proposes a novel integrated approach for the generation of such construction schedules considering the spatial aspect of construction management and in particular the temporal-space allocation of workspaces in site, proving that a potential and effective path may depend on the integration in a unique system architecture of three components: (1) BIM data (IFC-compliant) in terms of input-data source (2) generative algorithms for workspaces control, (3) artificial intelligence for generating construction schedules.

Based on the aforementioned components, this *PhD thesis* presents an expert-system based on a theoretical spatial-scheduling algorithm. It is supported by an ontology-based semantic modeling of the construction process knowledge. Therefore, a complete ontology, that was still missing in literature, was developed to formally represent the construction process management by using a multi-domain modelling approach which merges four sub-ontologies designated to formally describe the scheduling domain, time domain as well as workspaces and building products domains in terms of site entities, properties and relations among them.
Such an ontological framework was rendered into a *Protégé's script* in order to convert it in machine-readable language. According to this knowledge-base, a rule-engine was designed and included in the proposed model. It supports automated ontology-based reasoning mechanisms for schedules generation according to predefined planning rules.

This prototype introduces significant automation in existing scheduling processes, allowing both workspaces control mechanisms and construction schedule visualization.

Infact, as regards the **workspaces control**, the expert system incorporates an integrated model able to manage visualization, detection and resolution of workspaces conflicts. A 'bounding-box' concept was introduced to discretize geometries of both building objects and workspaces to rectangular shapes and it was integrated in a built-in algorithm, by means of a *Dynamo's script*, in order to automatically sculpt such geometries with minimized input work. A workspace conflicts detection and visualization process based on a BIM-based virtual environment was introduces in the system architecture. It provides a structured data package for the ontology-based rule-engine which in turn reschedules the conflicted activities together with the resource levelling.

The results of the validation test, conducted on an BIM-based project of an industrial building, showed that workspaces conflicts can be visually identified and automatically managed using the methodology and the system presented in this research.

The automated rescheduling functions, based on the ontological framework that drives the expert system in its entire, separate it from most of BIM studies on 4D simulation and technology that are focused in the simple visualization function of numerical construction data as opposed to the presented study. If those functions are classified by passive BIM systems (Moon et al., 2014), the proposed model can be classified as an *active BIM system*, thanks to the fact that it integrated a rule-based artificial intelligence to provide the shortest construction schedule (constraint-based) of the given Building Information Model.
By doing so, the 4D visualization environment is not considered the start-point of the scheduling process but it only constitutes the end-point for the construction schedule visualization by using all those data provided by the system that were, therefore, already been processed.

In terms of **workspaces planning**, the research introduces in the proposed system architecture the space syntax analysis which is used, so far, to understand environmental behavior mainly in healthcare settings and street networks. This can be considered the first application of this theory in the field of construction management for workspaces planning purposes.
By means of a *Grasshopper's script*, a workspaces allocation pattern for each construction method was automatically generated and then applied to each building object. This resulted in 'full bounding-box model' where each building object was simulated with the required workspaces for its installation. Moreover, the algorithm considered connectivity values among workspaces –established by the user's experience- as visualized in a connectivity map in generating the workspaces allocation pattern.

The obtained results show how the layout allocations of spaces cannot be considered never fixed but they strictly depend on both the chosen construction method and planning rules.

Infact, the space syntax analysis, carried out on six different construction methods, demonstrated that even if workspaces had the same dimensions, their layout allocations may be different depending on their relations as well as their interaction values.
The aforementioned remarks demonstrate how important should be the integration of human experiences even in advanced scheduling system and the pretty unbreakable bond that exists between a construction schedule and the spatial allocation of workspaces as well as their topological interactions. Infact, having perfectly the same building characteristics, if the construction methods change, the workspaces allocations changes as well as conflicts among

them that affect both the total construction time duration and temporal relations between activities.
Therefore, a predictive model aiming to support the schedules generation would have to consider this accuracy as this study made.

During the validation test, a particular attention was devoted to the **evaluation of construction alternatives** that generally emerge within both design and execution phase. Several solutions –construction sequences- that lead to the same construction may exist. The proposed system generated the construction sequence that ensured the shortest construction duration (constraint-based), which was not deductible a priori, and compared such a sequence with a different one (sequence bottom-up, left-to-right). In the figure below (Figure 12-1), the bar chart shows the comparison results. The schedule gets up from 36 construction days required to run out the shortest construction completion sequence, to 96 construction days when a different construction strategy was imposed. It is important to note that such construction alternatives completely include the same constructive demands in terms of both activities time-duration, resources availability and workspaces configuration pattern.
This comparison proves that for a given building project it cannot exist only one shortest construction sequence but it can exist under the same scheduling constraints mainly in terms of space requirements and planning rules, infact, the factor affecting this lag is that both schedules are placed under the scheduling constraint of non-overlapping workspaces.

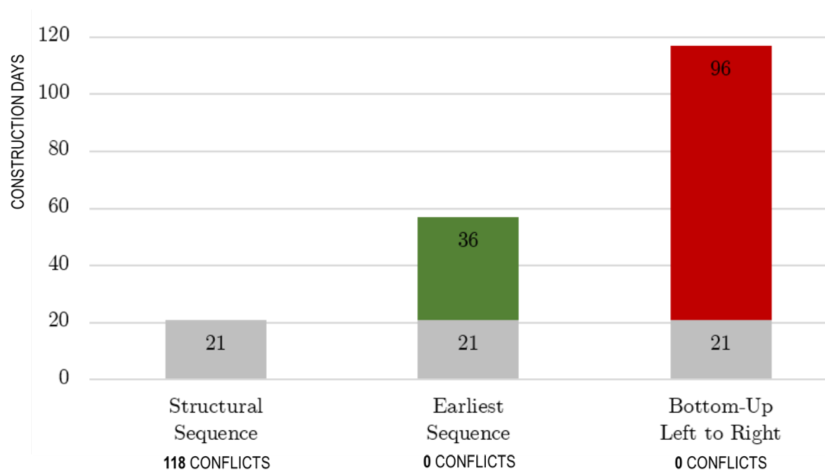


Figure 12-1 Bar Chart to compare the simulated construction schedules, as generated by the system

Concluding, the proposed expert system can be considered a precursor study in the direction of *BIM-based intelligent models* in the realm of construction management studies.
The ontology-based scheduling model, by means of the three-different but integrated scripts that underpin the system, supported the generation, optimization and management of site entities as well as relationships and dimensions of 611 workspaces by binding the scheduling process to such information. The same process would be unsustainable if manually managed or without a holistic and digitized planning process. This requirement increases if we think that

if some changes occur (e.g., construction methods, workspaces, building product dimensions, etc.) all the entities should be modelled again. The proposed model automates this processes and digests the changes, supporting efficiently the decision-making process.

12.2 Further developments

Further goals of future research and implementation are the following:

(1) **Expansion of the Knowledge Base**:

- Acquiring new domains-specific knowledge is considered one of the most powerful research extension. After all, the proposed KB did not follow an all-in-one modelling approach but analyzed the individual domains separately (i.e., scheduling, time, workspaces, building products). This choice of multi-ontologies paves the way to further developments of *new ontologies* in order to increase the model accuracy. Those ontologies may concern, for example, the *Health and Safety management and Risk assessment*. Such an expansion should be carried out concomitantly with new planning rules, to introduce within the rule-engine, providing the system the ability to generate a schedule that optimizes more than one variable at play identifying different types of overlapping conditions. For example, at the current status two activities cannot progress concurrently if their workspaces overlap (spatial conflicts) but, introducing new ontologies, two activities could not progress concurrently if their workspaces produce a too-high risk level (*safety conflicts*).

(2) **Incorporation of the Cost factor**:

- Adding the *cost dimension* is a future development for expanding the system simulation functions. This could be useful to refine the decision-making process. Infact, different construction alternatives could be compared not only by using the time dimension but also the cost dimension (e.g., cost of facilities, equipment, cost of relocations, etc.).

(3) **Improvement of workspaces management**:

- The individual workspaces are currently constructed based on the space dimensions provided by the user and then the control of the position of 3D shapes within the 3D space is carried out by using the space syntax analysis (Grasshopper's script) with reference to the others spaces and the building object. An *automatic control of workspaces dimension* with reference to the building object would be a model advancement. This would have a particular effect if we consider not punctual building objects as columns (e.g., walls, windows, ramps). It could be, for example, a direct assignment of the width of the workspace on the basis of the width of the building object.

- Moreover, at the current status, the construction site space was considered occupied by only workspaces referred to the installation of building objects. However, the identification and *incorporation of on-site paths* –taken between objects- in the space analysis need to be investigated to enable the system to reach more accuracy with regards to real site conditions.

(4) **Refinement of temporal scheduling properties** (durations and relations):

- The proposed model considers activities with a fixed duration. Considering '*contingency*', also known as 'buffer', which is an allowance specifically added to an activity to take account of unforeseen circumstances, could be the subject of further improvements.

- Furthermore, *advanced temporal relations* were not addressed. According to more advanced scheduling approaches, other types of relations between activities could be considered in the future (e.g., point-to-point, continuous relations).

(5) **Expansion of the expert system to the construction stage for the real-time workspaces monitoring**:

- The construction industry has a great interest in integrated systems able to monitor construction progress to check for deviation from the planned and agreed schedule during the design stage, in which the proposed model actually acts.

  At current status, knowing and analyzing the time-space allocation of construction resources (workspaces) the system provides a schedule that, in the future, could be used to monitor the execution phase in order to identify the status of activities referred to the schedule itself, improving construction productivity.

  Several *real-time control technologies* such as GPS, UWB, vision tracking or remote sensors technology can be implemented to collect 3D/4D (spatio-temporal) data. One of the mentioned tracking technologies could be selected to monitor workspaces status providing the system new data able to activate reasoning mechanisms to check deviation from the schedule.

  By expanding the proposed model, in the way it is structured by now, it would be able to check overlapping activities -due to *on-going changes of spaces availability*- and to reschedule the construction sequence saving the assumptions used to produce the first schedule. This would require a new integrated ontology to formally depict the monitoring domain and a rules package to activate control mechanisms.

References

Akbas, R. 2004. *Geometry-Based Modeling and Simulation of Construction Processes.* STANFORD UNIVERSITY, CIFE Technical Report.

Akinci, B., and J. Kunz, R. Levitt M. Fischer. 2000. "Representing Work Spaces Generically in Construction Method Models." *CIFE Working paper* (Stanford Univeristy).

Akinci, B., H. Karimi, A. Pradhan, C. Wu, and G. Fichtl. 2010. "CAD and GIS interoperability through semantic web services." *Journal of Information Technology in Construction* 13: 39–55.

Akinci, B., M. Fischer, and J. Kunz. 2000c. "Automated Generation of Work Spaces Required." Edited by Stanford Univeristy. *CIFE Working Paper.*

Allen, J., and G. Ferguson. 1997. *Actions and events in interval temporal logic.* ed., Kluwer O. Stock. http://dx.doi.org/10.1007/978-0-585-28322-7_7.

Atkinson, C., M. Gutheil, and K. Kiko. 2007. "On the Relationship of Ontologies and Models." Edited by Chair of Software Technology.

Azhar, S., M. Hein, and B. Sketo. 2008. "Building Information Modeling (BIM): benefits, risks and challenger." *Proceedings of the 44th ASC Annual Conference.* Alabama. 627-634.

Baader, F., D. Calvanese, D. Mcguiness, D. Nardi, and P., F. Patel-Schneider. 2003. *Description Logic Handbook: Theory, Implementations, and Applications.* Cambridge: Cambridge University Press.

Baader, F., D. Calvanese, D. McGuinness, P. Patel-Schneider, and D. Nardi. 203. *The Dscription Logic Handbook: Theory, Implementation and Applications.* Cambridge Univerisy Press.

Bargstädt, H., and A. Elmahdi. 2010. "Automatic generation of workspace requirements using qualitative and quantitative description." *Proceedings of the 10th International Conference on Construction Applications of Virtual Reality.* Japan: K. Makanae, N. Yabuki, K. Kashiyama.

Beetz, J., J. Van Leeuwen, and B. De Vries. 2009. "IfcOWL: A case of transforming EXPRESS schemas into ontologies." *Journal Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 23 (01): 89-101.

Benevolenskiy, A., K. Roos, P. Katranuschkov, and J. Scherer R. 2012. "Construction processes configuration using process patterns." *Advanced Engineering Informatics* (Elsiever) (26): 727-736.

Bordoli, A., and D. Baldwin. 2014. *A Handbook for Construction Planning and Scheduling.* West Sussex: John Wiley & Sons.

Buildingsmart. 2014. *Industry Foundation Classes Release 4 (IFC4).* http://www.buildingsmart-tech.org/ifc/IFC4/final/html/link/structural-connectivity.htm.

Cai, X., Su, H. 2014. "Life Cycle Approach to Construction Workspace Modeling and Planning." *Journal of Construnction Engineering and Management* (ASCE) March.

Chau, K., W., M. Anson, and J., P. Zhang. 2004. "Four-dimensional visualization of construction scheduling and site utilization." *Journal of Construction Engineering and Management* (ASCE) 130: 598–606.

Chavada, R., N. Dawood, and M. Kassem. 2012. "Construction workspace management: the development and application of a novel nD planning approach and tool." *Journal of Information Technology Construction (ITcon)* 12: 213–236.

Cheng, T., and J. Teizer. 2013. "Real-time resource location data collection and visualization technology for construction safety and activity monitoring applications." *Automation in Construction* 34: 3–15.

Cherneff, J., M., R. Logcher, and D. Sriram. 1991. "Integrating CAD with construction-schedule generation." *Journal of Computing in Civil Engineering* (ASCE) 5 (1): 68-84.

CIOB. 2011. *Guide to Good Practice in the Management of Time in Complex Projects.* Oxford: Wiley-Blackwell.

Connor, O., and D. Martin. 2009. *Importing Data into Protégé-OWL.* Stanford Center for Biomedical Informatics Research. http://protege.stanford.edu/conference/2009/slides/ImportingDataProtegeConference2009.pdf.

Cordier, J., M., and T. Porte. 1989. *Shape Theory: Categorical Methods of Approximation.* New York: Dover publications.

Cox, S., and C. Little. 2016. "Time Ontology in OWL." *http://www.w3.org/2006/time.*

Darwiche, A., R. Levitt, and B. Hayes-Roth. 1988. "OARPLAN: Generating Project Plans by Reasoning about Objects, Actions and Resources." *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 2 (3): 169-181.

Dawood, N., and S. Sikka. 2009. "Development of 4D based performance indicators in construction industry." *Engineering Construction and Architectural Management* (Wiley) 16: 438–458.

Dawood, N., and Z. Mallasi. 2006. "Construction workspace planning: assignment and analysis utilizing 4D visualization technology,." *Computer-Aided Civil and Infrastructure Engineering* (Elseiver) 21 (7): 498–513.

Dawood, N., D. Scott, E. Sriprasert, and Z. Mallasi. 2005. "The virtual construction site (VIRCON) tools: An industrial evaluation." *Journal of Information Technology in Construction (Special Issue: From 3D to nD modelling)* 10: 43-54.

Dhillon, R., K., M. Jethwa, and H., S. Rai. 2014. "Extracting Building Data from BIM with IFC." *International Journal on Recent Trends in Engineering and Technology* 11 (1).

Drogemulle, H., and R. Schevers. 2005. "Converting the Industry Foundation Classes to the Web Ontology Language."

Dudáš, M., O. Zamazal, and V. Svátek. 2014. "Roadmapping and nav- igating in the ontology visualization landscape." Edited by Springer. *Knowledge Engineering and Knowledge Management* (K. Janowicz, S. Schlobach, P. Lambrix, and E. Hyvönen) 8876: 137–152.

Dzeng, R., J., and I., E. Tommelein. 1997. "Boiler erection scheduling using product models and case-based reasoning,." *Journal of Construction Engineering and Management* (ASCE) 123 (3).

Echeverry, D. 1991. "Factors for generating initial construction schedules." Department of Civil Engineering, University of Illinois, PhD thesis, Urbana-Champaign.

Elbeltadi, E., T. Hegazy, and A. Eldosouky. 2004. "Dynamic layout of construction temporary faciltities considering safety." *Journal of Construction Engineering and Management* 130 (4): 534-541.

El-Diraby. 2013. "Domain ontology for construction knowledge." *Journal of Construction Engineering and Management* (ASCE) 768–784.

El-Diraby, T., and h. Osman. 2011. "Automation in Construction." *A domain ontology for construction concepts in urban infrastructure products* 20 (8): 1120–1132.

Ernstrom, B., and et al. 2006. *The contractors' Guide to BIM.* Associated General Contractors of America.

Ferguson, J., Allen. 1997. "Actions and Event in interval temporal logic." http://dx.doi.org/10.1007/978-0-585-28322-7_7.

Fortineau, V., T. Paviot, L. Louis-Sidney, and S. Lamouri. 2012. "SWRL as a rule language for ontology-based models in power plant design."

Friedman, E., H. 2003. *Rule-Based Systems in Java - JESS IN ACTION.* Greenwich: Manning Publications.

Fu, B., N., F. Noy, and M., A. Storey. 2013. "Indented tree or graph? a usability study of ontology visualization techniques in the context of class mapping evaluation." Edited by Springer. *The Semantic Web* (H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J. Parreira, L. Aroyo, N. Noy, C. Welty, and K. Janowicz) 8218: 117–134.

Gambatese, J., A., M. Behm, and J., W. Hinze. 2005. "Vviability of designing for construction worker safety." *Journal of Costruction Engineering and Management* (ASCE) 131 (9): 1029-1036.

Grosso, W., E., H. Eriksson, R., W. Fergerson, J., H. Gennari, S., W. Tu, and M., A. Musen. 1999. "Knowledge modeling at the millennium: The design and evolution of Protege-2000." Technical Report, Stanford University, Institute for Medical Informatics, Standford.

Gruber, T., R. 1995. "Toward principles for the design of ontologies used for knowledge sharing?" Edited by Elseiver. *International Journal of Human-Computer Studies* 43 (5): 907–928.

Gu, N., and K. London. 2010. "Understanding and facilitating BIM adoption in the AEC industry." *Automation in Construction* 19: 988–999.

Guo, H., H. Li, G. Chan, and M. Skitmore. 2012. "Using game technologies to improve the safety of construction plant operations." *Accident Analysis and Prevention* 48: 204–213.

Guo, S., J. 2002. "Identification and resolution of work space conflicts in building construction." *Journal of Construction Engineering and Management* (ASCE) 128: 287–295.

Guo, S., S., and C., W. Chan. 2011. "A comparison and analysis of some ontology visualization tools." Edited by SEKE. *Proceedings of the 23rd International Conference on Software Engineering & Knowledge Engineering.* Knowledge Systems Institute Graduate School. 357–362.

Hanson, B., and J. Hillier. 1984. *The Social logica of space.* Cambrige University Press.

Hegazy, T., M., and E. Elbeltagi. 1999. "EvoSite: Evolution-based models for site layout planning." *Journal of Computing in Civil Engineering* (ASCE) 13 (3): 198–206.

Heijst, G., A., Th. Schreiber, and B., J. Wielingua. 1997. "Using Explicit Ontolgies in KBS Development." *International Journal of Human-Computer Studies* (Academic Press) 46: 183-292.

Hillier, B. 2007. *Space is the Machine.* London: Cambridge University Press.

Hori, M., Y. Nakamura, H. Satoh, K. Maruyama, T. Hama, S. Honda, T. Takenaka, and F. Sekine. 1995. "Knowledge-level analysis for elicting composable scheduling knowledge." *Artificial Intelligenxe in engineering* (Elsiver Science Limited) 9: 253-264.

Horridge, M. 2011. "A Practical Guide To Building OWL Ontologies Using Prot́eǵe 4 and CO-ODE Tools Edition 1.3." Guide, The University Of Manchester.

Horrocks, I. 2004. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML.* https://www.w3.org/Submission/SWRL/.

Horrocks, I., P., F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. 2004. *SWRL: A semantic web rule language combining OWL and RuleML.*

IDLab, Ghent Univeristy's. 2013. *IFC-to-RDF converter.* IDLab. https://github.com/IDLabResearch/IFC-to-RDF-converter/wiki/IFC-to-RDF.

Jagbeck, A. 1994. "MDA Planner: Interactive Planning Tool Using Product Models and Construction Methods." *Journal of Computing in Civil Engineering* 8 (4): 536-554.

Kalinowski, K. 2012. "Multistage decision making process of multicriteria production scheduling." *Journal of Machine Engineering* 12 (3).

Kano, N. 1990. "A knowledge-based system for construction planning and scheduling: a prototype system based on the down-from the-top methodology." *The 7th International Symposium on Automation and Robotics in Construction.* Bristol. 303-311.

Kartam, N., A., and R., E. Levitt. 1990. "Intelligent planning of construction projects." *Journal of Computing in Civil Engineering* (ASCE) 4 (2): 155-176.

Kassem, M., N. Dawood, and R. Chavada. 2015. "Construction workspace management within an Industry Foundation Class-Compliant 4D tool." *Automation in Construction* 52: 42-58.

Katifori, A., C. Halatsis, G. Lepouras, A. Vassilakis, and E. Giannopoulou. 2007. "Ontology visualization methods – a survey." *ACM Computer Survey* 39 (4).

Katifori, A., E. Torou, C. Halatsis, G. Lepouras, and Vassilakis C. 2006. "A comparative study of four ontology visualization techniques in Protégé: Experiment setup and preliminary results." *International Conference on Information Visualisation.* IEEE. 417–423.

Kim Jongeling, J., M. Fischer, C. Mourgues, and T. Olofsson. 2008. "Temporary Structure Usage, and Productivity Using 4D Models." *Automation in Construction* 17: 780-791.

Konig, M., and A. Marx. 2013. "Modeling and simulating spatial requiremts of construction activities." *Winter simulation conference 2013.* R. Pasupathy et al.

Krishnamoorthy, C., S., and S. Rajeev. 1996. *Artificial Intelligence and Expert Systems for Engineers.* CRC Press LLC.

Lambert, D., and C. Nowak. 2008. "The Mephisto Conceptual Framework" *Command, Control, Communications and Intelligence Division DSTO Defence Science and Technology Organization.* Australia.

Lanzenberger, M., J. Sampson, and M. Rester. 2010. "Visualization in ontology tools." *Proceedings of the International Conference on Complex, Intelligent and Software Intensive Systems.* IEEE. 705–711.

Lee, K., J., H., W. Kim, J., K. Lee, and T., H. Kim. 1998. *Case and constraint-based project planning for apartment construction.* Vol. 19. 1 vols. AI Magazine.

Liao, S.H. 2005. "Expert System methodologies and applications - a decade review from 1995 to 2004." *Expert Systems with Applications* (Elsiever) (28): 93-103.

Lima, C., T. El-Diraby, and J. Stephens. 2005. "Ontology-based optimi- sation of knowledge management in e-construction." *Journal of Information Technology in Construction* 10: 305–327.

Little, S., and C. Cox. 2016. *Time Ontolgy in OWL.* http://www.w3.org/TR/2016/WD-owl-time-

20160712/.

Lohmann, S., S. Negru, F. Haag, and E. Ertl. 2014. "Visualizing Ontologies with VOWL." *Semantic Web* (IOS Press).

Lopez, M., F., A., G. Pérez, and N. Juristo. 1997. "Methontology: From ontological art towards ontological engineering." *Ontological Engineering Spring Symp. Series.* American Asso- ciation for Artificial Intelligence. 33-40.

Ma, Z., Q. Shen, and J. Zhang. 2005. "Application of 4D for dynamic site layout planning and management of construction projects." *Automation in Construction* 14: 369-381.

Mallasi, Z. 2006. "Dynamic quantification and analysis of the construction workspace congestion utilising 4D visualisation." *Automation in Construction* (Elseiver) (15): 640–655.

McGuinness, N., F. 2001. "Ontology Development 101: A Guide to Creating Your Fisrt Ontology." Standford Univerisity, Standford.

Mikulakova, E., M. Konig, E. Tauscher, and Beucke. K. 2010. "Knowledge-based schedule geenration and evaluation." *Advanced engineering Informatics* 24: 389-403.

Mohan, S. 1990. "Expert systems applications in construction management and engineering." *Journal of Construction Engineering and Management* (ASCE) 116 (1): 87-99.

Moon, H., H. Dawood, and L. Kang. 2014. "Development of workspace conflict visualization system using 4D object of work schedule." *Advanced Engineering Informatics* (Elseiver) 28: 50–65.

Moon, H., H. Kim, C. Kim, and L. Kang. 2014. "Development of a schedule workspace interference management system simultaneously considering the overlap level of parallel schedules and workspaces." *Automation in Construction* 39: 93–105.

Moon, H., H. Kim, V. R. Kamat, and Kang L. 2015. "BIM-based Construction Scheduling Method Using Optimization Theory for Reducing Activity Overlaps." *Journal of Computing in Civil Engineering* (ASCE) 29 (3).

Moon, H., L. Kang, N. Dawood, and S. Ji. 2009. "Configuration method of health and safety rule for improving productivity in construction space by multi-dimension CAD system." *Proceedings of ICCEM/ICCPM.* Korea. 27-30.

Morad, A., A., and Y., J. Beliveau. 1991. "Knowledge-based planning system." *Journal of Construction Engineering and Management* (ASCE) 117 (1): 1-12.

Motawa, I., and A., Almarshad. 2012. "A Knowledge-based BIM system for building maitenance." *Automation in Construction* (Elseiver) 29: 173-182.

Motta, E. 2000. "The knowledge modelling paradigm in knowledge engineering." *Handbook of Software Engineering and Knowledge Engineering* (World Scientific Publishing) 1: 1-29.

Mun, D., and K. Ramani. 2011. "Knowledge-based part similarity measurement utilizing ontology and multi-criteria decision making technique." *Advanced Engineering Informatics* 119-130.

Muna, D., and K. Ramanib. 2011. "Knowledge-based part similarity measurement utilizing ontology and multi-criteria decision making technique." *Advanced Engineering Informatics* 25 (2): 199-130.

Navinchandra, D., D. Sriram, and R. Logcher. 1988. "GHOST: project network generator." *Journal of Computing in Civil Engineering* (ASCE) 2 (3): 239-254.

Nourian, P., S. Rezvani, and S. Sariyildiz. 2013. "Designing with Space Syntax A con gurative approach to architectural layout, proposing a computational methodology." *eCAADe 31.* 357-365.

Noy, N., R., W. Fergerson, and M., A. Musen. 2000. "The knowledge model of Protege-2000: Combining interoperability and flexibility." *Proceedings of the 12th international conference on knowledge engineering and knowledge management.* France.

OpenBIM, BuildingSMART. 2016. *ifc-overview.* http://www.buildingsmart-tech.org/specifications/ifc-overview.

Rajpathak, D. 2001. "The Task Ontology Component of the Scheduling Library." *Knowledge Media Institute* (The open Univeristy). Pilot Study.

Rajpathak, D., E. Motta, and R. Roy. 2000. "A Generic Task Ontolgy for Scheduling Applications."

Riley, D., R., and V. Sanvido. 1997. "Space planning method for multistory building construction." *Journal of Construction Engineering and Management* (ASCE) 171–180: 171–180.

Sacks, R., O. Rozenfeld, and Y. Rosenfeld. 2009. "Spatial and Temporal Exposure to Safety Hazards in Construction." *Journal of Cosntruciton Engineering and Management* (ASCE) 135 (8): 726-736.

Said, H., and G. Lucko. 2016. "Float Types in Construction Spatial Scheduling." *Journal of Construction Engineering and Management* (ASCE) 142 (12): 1-12.

Sanvido, D. R. Riley and V. E. 1995. "Patterns of Construction-Space Use in Multistory Buildings." *J. Constr. Engine. Manage.* 123 (4): 464-473.

Sayed, K., A. Turner, B. Hillier, S. Iida, and A. Penn. 2014. *Space syntax methodologies.* Edited by Bartlett School of Engineering Architecture. London.

Shaked, O., and A. Warszawski. 1995. "Knowledge based system for construction planning of high-rise buildings." *Journal of Construction Engineering and Management* (ASCE) 110 (2): 172-182.

Smith, S., F., and M., A. Becker. 1997. "An Ontolgy of Constructing Scheduling Systems." *AAAI Symposium on Ontological Engineering.* Stanford: AAAI Press.

Smith, S., F., G. Cortellessa, D., W. Hildum, and D., W. Ohler. 2005. "Using a scheduling domanin ontolgy to compute user-oriented explanations." *Planning, Scheduling, and Constraint Satisfaction: From theory to Practise.*

Song, Y., and D. Chua. 2005. "Detection of spatio-temporal conflicts on a temporal 3D space system." *Advanced Engineering Software* (36): 814–826.

Steuer, J. 1993. "Defining virtual reality: dimensions determining telepresence." *Journal of Communication* 42: 73–93.

Su, X., R. Andohar, J. Pan, and A. Kandil. 2012. "GIS-based dynamic construction site material layout evaluation for building renovation projects." *Automation in Construciton* 27: 40-49.

Tah, J., H., M., V. Carr, and R. Howes. 1999. "Information modeling for case-based construction planning of highway bridge proj- ects." *Advances in Engineering Software* (Elseiver) 30 (7): 495-509.

Thabet, W., Y., and Y., J. Beliveau. 1994. "Modeling work space to schedule repetitive floors in multistory buildings." *Journal of Construction Engineering and Management* 120 (1): 96–116.

Thomas, H., R., D., R. Riley, and S., K. Sinha. 2006. "Fundamental principles for avoiding congested work areas - a case study." *Practice Periodical on Structural Design and Construction* (11): 197–205.

Wang, H., and F. Boukamp. 2011. "Ontology-based representation and reasoning framework for supporting job hazard analysis." *Journal of Computing in Civil Engineering* (ASCE) 25 (6): 442–456.

Wikipedia. 2017. *Industry Foundation Classes.* https://en.wikipedia.org/wiki/Industry_Foundation_Classes.

Winch, G., M., and S. North. 2006. "Critical space analysis." *Journal of Construcition Engineering and Management* 132: 473–481.

Zhang, S., F. Boukamp, and J. Teizer. 2015. "Ontology-based semantic modeling of construction safety knoledge: Towards automated safety planning for job hazard analsis." (Automation in Construction) 52.

Zhang, S., F. Boukamp, and J. Teizer. 2015. "Ontology-based semantic modeling of construction safety knowledge: towards automated safety planning for job hazard analysis (JHA)." *Automation in Construction* (Elseiver) 52: 29–41.

Zhang, S., J. Teizer, N. Pradhananga, and C., E. Eastman. 2015. "Workforce location tracking to model, visualize and analyze workspace requirements in building information models for construction safety planning." *Automation in Construction* 60: 74-86.

Zhong, B., L. Ding, H. Luo, Y. Zhou, Y. Hu, and H. Hu. 2012. "Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking." *Automation in Construction* (Elsevier) (28): 58–70.

Zhou, Z., Y., M. Goh, and L. Shen. 2016. "Overview and Analysis of Ontology Studies Supporting Development of the Construction Industry." *J.ournal of Computing in Civil Engineering* (ASCE).

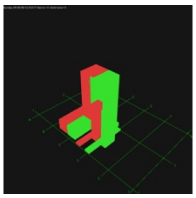
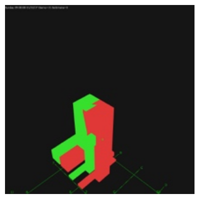
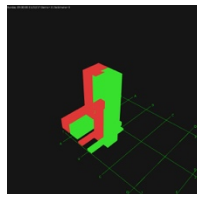
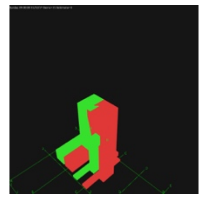
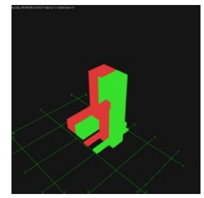
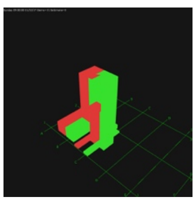
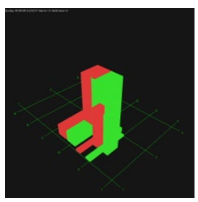
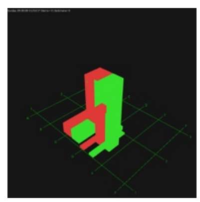
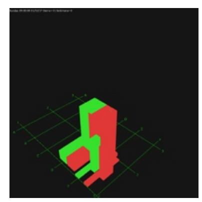
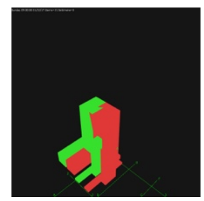
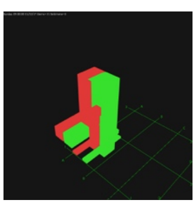
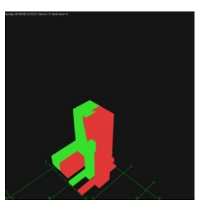
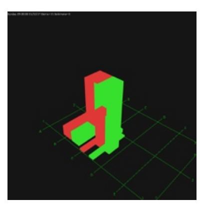
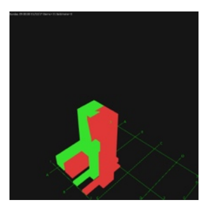
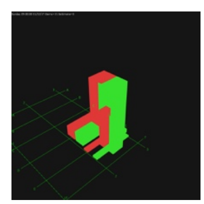
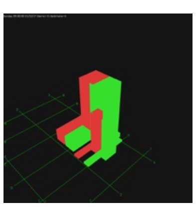
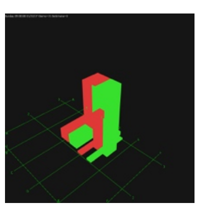
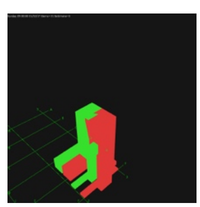
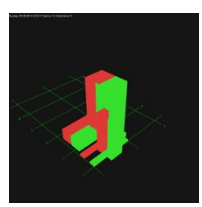
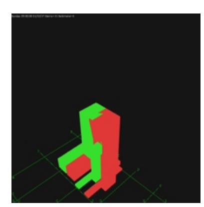
Zouein, P., P., and I., D. Tommelein. 1999. "Dynamic layout planning using a hybrid incremental solution method." *Journal of Construction Engineering and Management* 125 (6): 400-408.

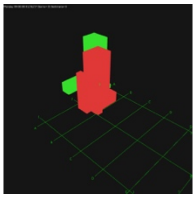
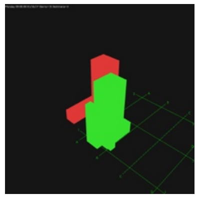
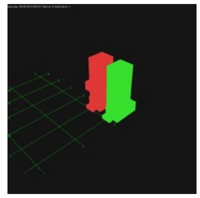
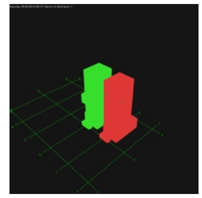
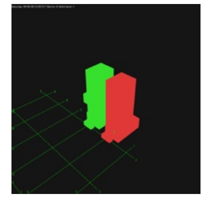
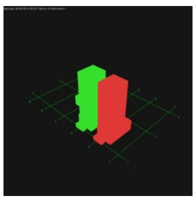
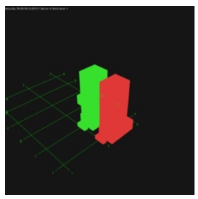
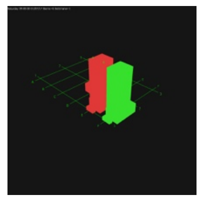
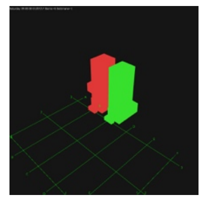
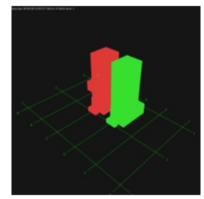
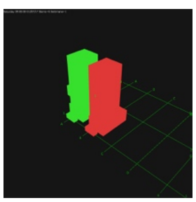
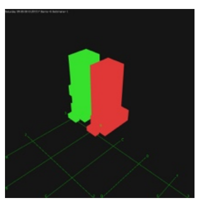
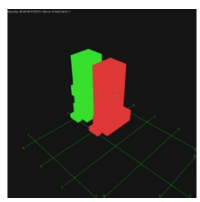
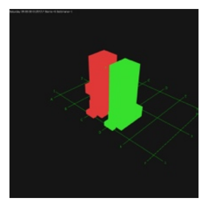
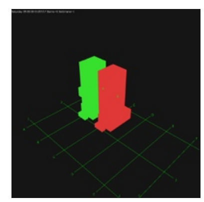
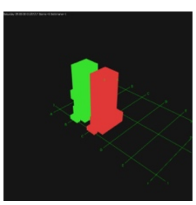
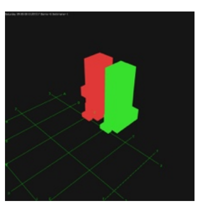
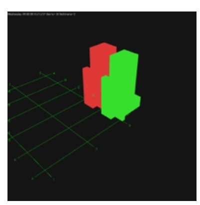
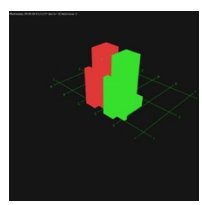
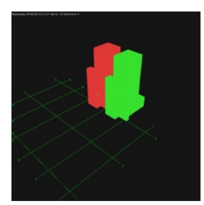
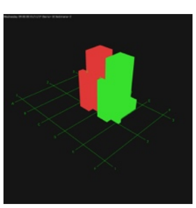
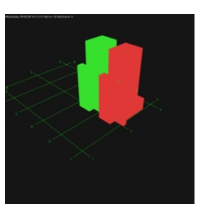
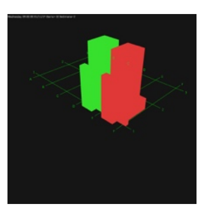
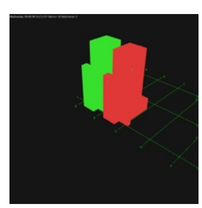
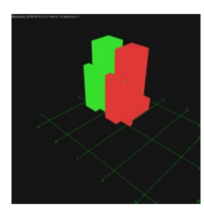
Zozaya-Gorostiza, C., C. Hendrickson, and D., R. Rehak. 1989. *Knowledge-Based Process Planning for Construction and Manufacturing.* Academic Press.

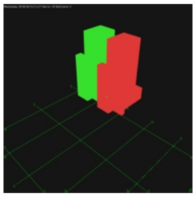
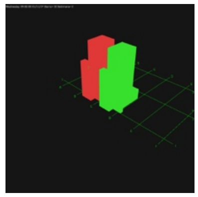
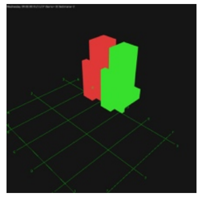
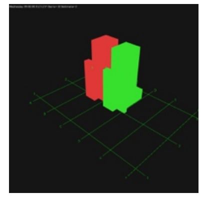
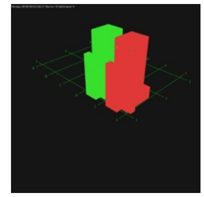
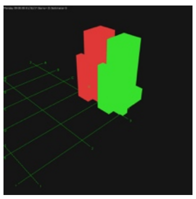
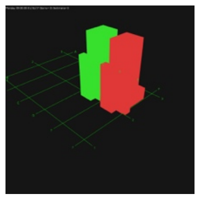
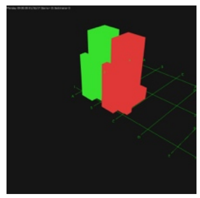
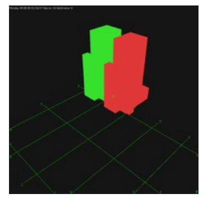
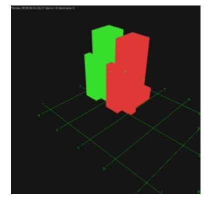
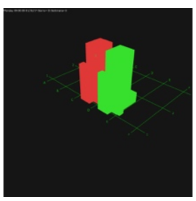
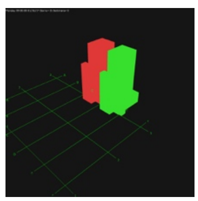
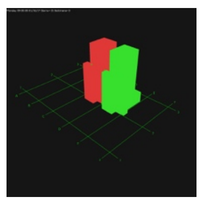
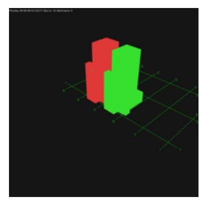
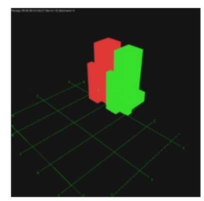
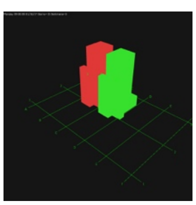
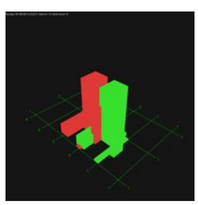
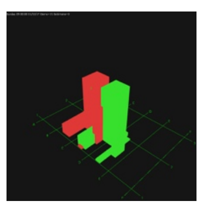
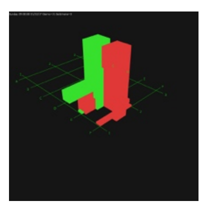
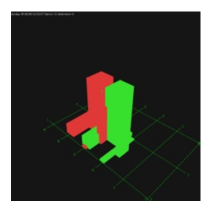
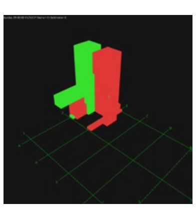
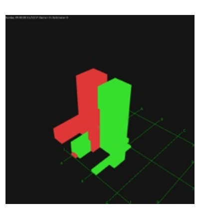
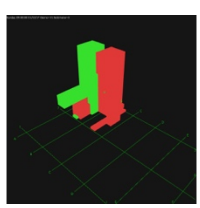
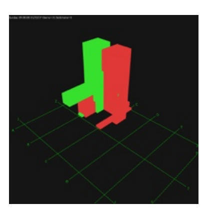
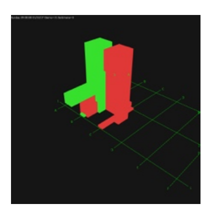
Annex (1)

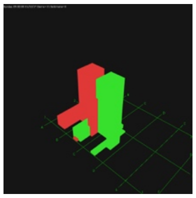
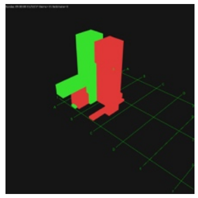
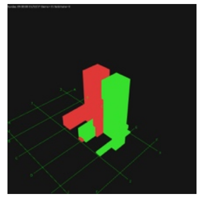
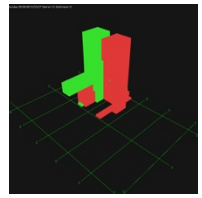
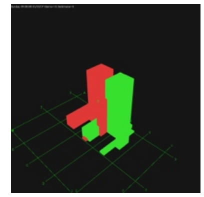
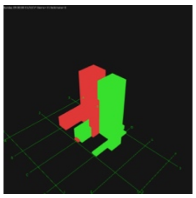
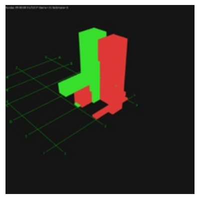
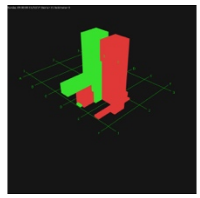
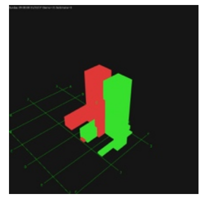
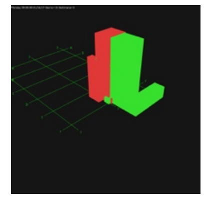
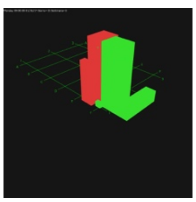
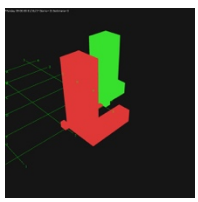
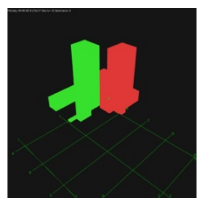
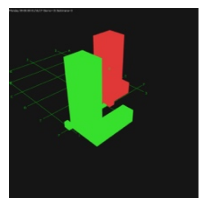
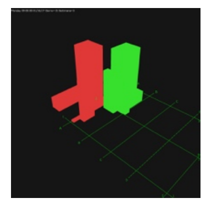
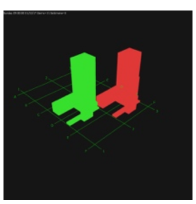
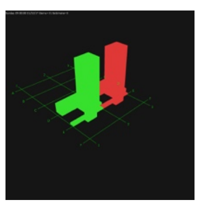
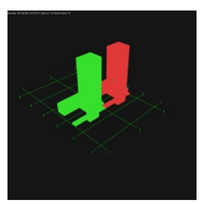
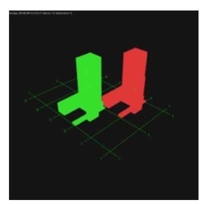
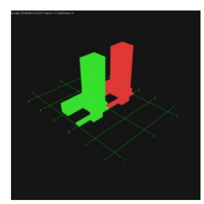
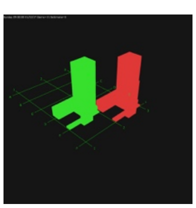
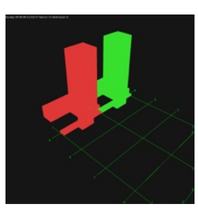
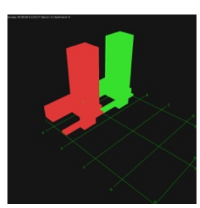
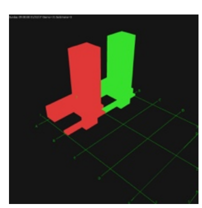
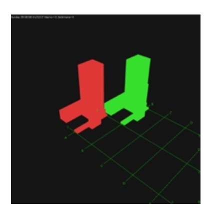
Output data of the **workspaces conflicts checking process.**

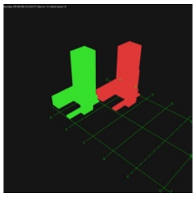
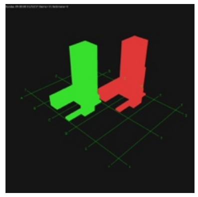
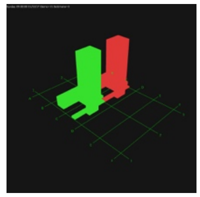
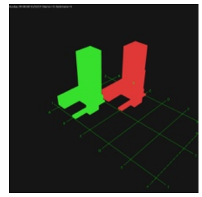
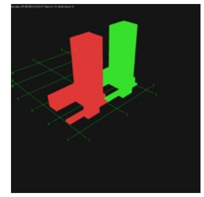
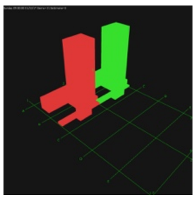
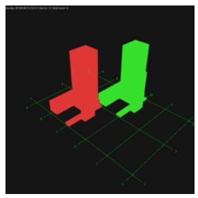
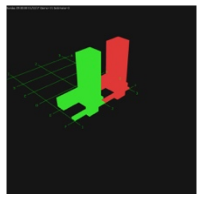
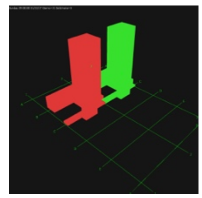
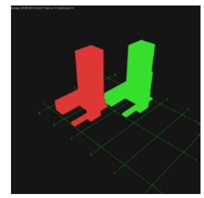
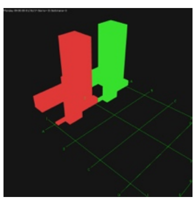
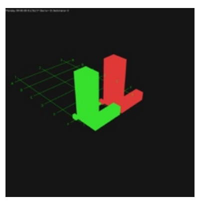
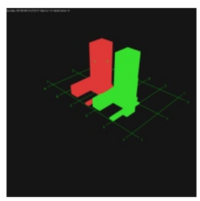
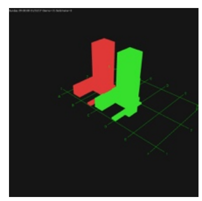
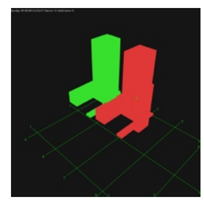
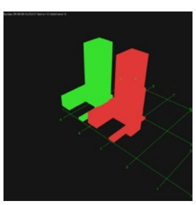
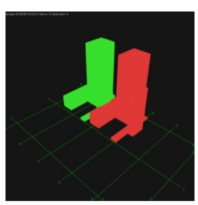
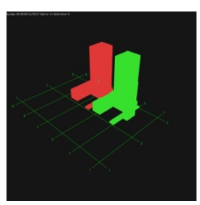
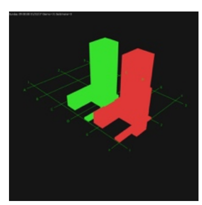
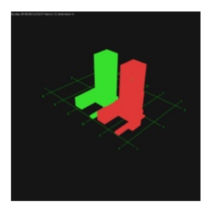
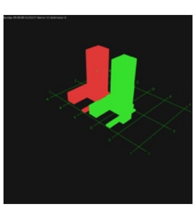
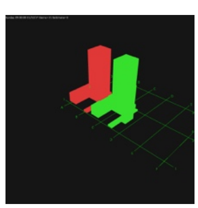
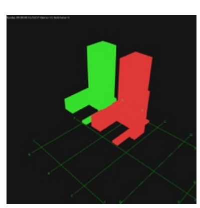
The report tab includes: `Conflict ID`, `Graphical representation`, `Grid Location` [*meters*].

1	**2**	**3**	**4**	**5**
				
x:32.866, y:12.203, z:17.600	x:7.506, y:39.123, z:26.000	x:12.366, y:12.203, z:17.600	x:18.146, y:39.123, z:26.000	x:32.866, y:30.823, z:17.600
6	**7**	**8**	**9**	**10**
				
x:23.006, y:12.203, z:17.600	x:43.506, y:12.203, z:17.600	x:38.186, y:12.203, z:17.600	x:44.746, y:20.503, z:26.000	x:2.186, y:39.123, z:26.000
11	**12**	**13**	**14**	**15**
				
x:7.046, y:12.203, z:17.600	x:12.826, y:39.123, z:26.000	x:27.546, y:12.203, z:17.600	x:12.826, y:20.503, z:26.000	x:48.826, y:30.823, z:17.600
16	**17**	**18**	**19**	**20**
				
x:43.506, y:30.823, z:17.600	x:38.186, y:30.823, z:17.600	x:50.066, y:39.123, z:26.000	x:54.146, y:12.203, z:17.600	x:23.466, y:39.123, z:26.000

21	22	23	24	25
				
x:8.171, y:28.763, z:0.000	x:8.171, y:5.663, z:17.600	x:46.871, y:48.713, z:0.000	x:33.971, y:34.913, z:5.000	x:33.971, y:53.413, z:5.000
26	**27**	**28**	**29**	**30**
				
x:33.971, y:16.413, z:5.000	x:44.971, y:34.913, z:5.000	x:46.871, y:11.713, z:0.000	x:13.871, y:48.713, z:0.000	x:24.871, y:30.213, z:0.000
31	**32**	**33**	**34**	**35**
				
x:24.871, y:11.713, z:0.000	x:9.871, y:30.213, z:5.000	x:9.871, y:48.713, z:5.000	x:9.871, y:11.713, z:5.000	x:20.871, y:30.213, z:5.000
36	**37**	**38**	**39**	**40**
				
x:20.871, y:11.713, z:5.000	x:24.871, y:48.713, z:0.000	x:48.171, y:39.363, z:1.000	x:37.171, y:2.363, z:1.000	x:37.171, y:39.363, z:1.000
41	**42**	**43**	**44**	**45**
				
x:37.171, y:20.863, z:1.000	x:48.171, y:20.863, z:1.000	x:48.171, y:2.363, z:1.000	x:15.171, y:2.363, z:1.000	x:15.171, y:20.863, z:1.000

46	47	48	49	50
 x:15.171, y:39.363, z:1.000	 x:26.171, y:2.363, z:1.000	 x:26.171, y:39.363, z:1.000	 x:26.171, y:20.863, z:1.000	 x:48.171, y:2.763, z:0.600
51	**52**	**53**	**54**	**55**
 x:48.171, y:38.763, z:0.600	 x:48.171, y:20.163, z:0.600	 x:15.171, y:2.463, z:0.600	 x:15.171, y:38.463, z:0.600	 x:15.171, y:20.163, z:0.600
56	**57**	**58**	**59**	**60**
 x:37.171, y:2.763, z:0.600	 x:37.171, y:38.763, z:0.600	 x:37.171, y:20.163, z:0.600	 x:26.171, y:2.463, z:0.600	 x:26.171, y:38.463, z:0.600
61	**62**	**63**	**64**	**65**
 x:26.171, y:20.163, z:0.600	 x:39.705, y:12.203, z:17.600	 x:34.385, y:12.203, z:17.600	 x:51.126, y:12.203, z:0.000	 x:29.065, y:12.203, z:17.600
66	**67**	**68**	**69**	**70**
 x:8.566, y:32.723, z:0.000	 x:8.566, y:12.203, z:17.600	 x:13.886, y:32.723, z:0.000	 x:19.206, y:30.823, z:0.000	 x:19.206, y:12.203, z:0.000

71	**72**	**73**	**74**	**75**
				
x:24.526, y:12.203, z:17.600	x:13.886, y:14.103, z:0.000	x:39.705, y:30.823, z:17.600	x:24.526, y:30.823, z:0.000	x:34.385, y:30.823, z:17.600
76	**77**	**78**	**79**	**80**
				
x:29.065, y:30.823, z:17.600	x:51.126, y:30.823, z:0.000	x:45.806, y:12.203, z:0.000	x:45.025, y:30.823, z:17.600	x:57.643, y:20.421, z:0.600
81	**82**	**83**	**84**	**85**
				
x:57.643, y:2.563, z:0.600	x:57.643, y:38.663, z:0.600	x:5.699, y:38.463, z:0.600	x:57.643, y:20.163, z:0.600	x:5.699, y:20.163, z:0.600
86	**87**	**88**	**89**	**90**
				
x:41.106, y:22.003, z:0.000	x:46.426, y:20.723, z:0.840	x:41.106, y:20.723, z:0.840	x:35.786, y:22.003, z:0.000	x:35.786, y:20.723, z:0.840
91	**52**	**93**	**94**	**95**
				
x:46.426, y:22.003, z:0.000	x:3.866, y:22.003, z:0.000	x:9.186, y:22.003, z:0.000	x:14.506, y:22.003, z:0.000	x:3.866, y:21.723, z:2.000

96	97	98	99	100
				
x:9.186, y:22.003, z:0.000	x:30.466, y:22.003, z:0.000	x:30.466, y:20.723, z:0.840	x:14.506, y:22.003, z:0.000	x:51.746, y:22.003, z:0.000
101	**102**	**103**	**104**	**105**
				
x:19.826, y:22.003, z:0.000	x:25.146, y:21.723, z:2.000	x:57.066, y:20.723, z:0.840	x:25.146, y:22.003, z:0.000	x:19.826, y:21.723, z:2.000
106	**107**	**108**	**109**	**110**
				
x:5.699, y:21.663, z:0.000	x:57.643, y:20.163, z:0.000	x:35.166, y:8.503, z:0.000	x:24.526, y:8.503, z:0.000	x:13.446, y:27.123, z:5.000
111	**112**	**113**	**114**	**115**
				
x:13.446, y:8.503, z:5.000	x:24.086, y:27.123, z:5.000	x:35.166, y:27.123, z:0.000	x:45.366, y:8.503, z:5.000	x:40.046, y:8.503, z:5.000
116	**117**	**118**		
				
x:29.846, y:8.503, z:0.000	x:19.206, y:8.503, z:0.000	x:8.126, y:27.123, z:5.000		

An ontology-based BIM Expert System for temporal and spatial construction planning

2018, Vito Getuli
International Doctorate in Civil and Environmental Engineering

Supervisors: Prof. Klaus Thiele and Prof. Pietro Capone

Reviewers: Prof. Shabtai Isaac and Prof. Angelo Luigi Camillo Ciribini